

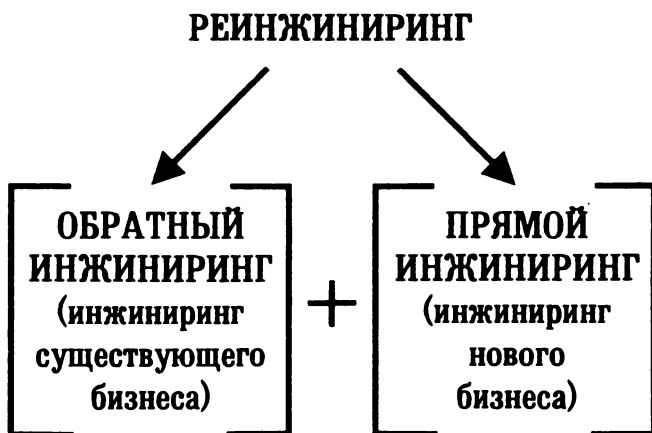
Е.Г. Ойхман, Э.В. Попов

РЕИНЖИНИРИНГ БИЗНЕСА



ИНЖИНИРИНГ-
от английского слова
engineering -
сооружать, проектировать, устраивать,
затевать, придумывать,
изобретать

РЕИНЖИНИРИНГ,
или, точнее,
БИЗНЕС-ПРОЦЕСС РЕИНЖИНИРИНГА (БПР),
обозначает совокупность методов и средств,
предназначенных для кардинального
улучшения основных показателей деятельности
компании (предприятия) путем анализа
и перепроектирования существующих
бизнес-процессов



Е.Г. Ойхман, Э.В. Попов

РЕИНЖИНИРИНГ БИЗНЕСА :

**реинжиниринг
организаций
и информационные
технологии**



МОСКВА
"ФИНАНСЫ И СТАТИСТИКА"
1997

УДК 681.518:339.13
ББК 65.200-2с
О-48

Рецензент: Председатель Совета Российской
Ассоциации искусственного интеллекта
доктор технических наук,
академик РАЕН *Д.А. Поспелов*

Издание осуществлено при поддержке
Российского фонда фундаментальных исследований

Ойхман Е.Г., Попов Э.В.

О-48 **Реинжиниринг бизнеса: Реинжиниринг организаций и
информационные технологии. - М.: Финансы и статистика,
1997. - 336 с.: ил.**

ISBN 5-279-01791-4.

Книга посвящена исследованию достижений бизнес-процесса реинжиниринга (БПР), принятого на вооружение ведущими компаниями мира. Показана ведущая роль информационных технологий в реинжиниринге бизнес-процессов. Дана общая методология проведения БПР на основе интегрированного подхода. Описаны инструментальные средства поддержки БПР.

Для менеджеров, разработчиков проектов по БПР, разработчиков информационных систем, а также преподавателей, аспирантов и студентов вузов соответствующих специальностей.

2404000000 - 034

О 112 - 97
010(01) - 97

ББК 65.200-2с

ISBN 5-279-01791-4

© Е.Г Ойхман, Э.В Попов, 1997



Понятие *бизнес-процесс реинжиниринг* (БПР - Business process reengineering) возникло примерно в 1990 г. и с тех пор вызывает активный интерес специалистов в области менеджмента и информационных технологий. С 1994 г. в США проводятся ежегодные конференции по БПР. Издано более десятка монографий и сотни статей, описывающих БПР. Наибольший успех выпал на книгу М. Хаммера и Дж. Чампи "Реинжиниринг корпораций: революция в бизнесе" (Hammer M. and Champy J. Reengineering the Corporation: A Manifesto to Business Revolution), изданную в 1993 г.: в течение двух лет она была самой продаваемой не только среди научно-технических, но и среди художественных книг.

В настоящее время БПР взят на вооружение почти всеми ведущими компаниями мира. В частности, по данным фирмы Ernst & Young, 100 крупнейших банков Северной Америки затратят в 1997 г. около 2,9 млрд. дол. на реинжиниринг своих подразделений. За последние полтора года правительство США начало более 200 проектов по реинжинирингу.

М.Хаммер и Дж.Чампи определяют реинжиниринг как "фундаментальное переосмысление и радикальное перепроектирование бизнес-процессов компаний для достижения коренных улучшений в основных актуальных показателях их деятельности - стоимость, качество, услуги и темпы". Имеется в виду не небольшое усовершенствование бизнес-процессов компаний - например, на 10-100 %, а кардинальное повышение их эффективности - в десятки раз.

Необходимость реинжиниринга обосновывается высокой динамичностью современного делового мира. Непрерывные и довольно существенные изменения в технологиях, рынках сбыта и потребностях клиентов стали обычным явлением, и компании, стремясь выжить и сохранить конкурентоспособность, вынуждены непрерывно перестраивать свою стратегию и тактику.

Решением проблемы является смена базовых принципов организации компаний и переход к ориентации не на функции, а на процессы. Из всех концепций менеджмента, основанных на процессах, БПР рассматривается как наиболее эффективная концепция. М.Хаммер, автор термина "реинжиниринг", считает появление БПР революцией

в бизнесе, которая знаменует отход от базовых принципов построения компаний, предложенных 200 лет назад А.Смитом, и превращает конструирование бизнеса в инженерную деятельность. Возможность такой революции обусловлена в первую очередь новейшими достижениями в области информационных технологий, в частности, технологии динамических экспертных систем.

Важность БПР возрастает в связи с тем, что, во-первых, он “беспартиен”, т.е. он необходим любой компании, вне зависимости от рода ее деятельности, и, во-вторых, БПР необходим для больших, средних и даже маленьких компаний. Как говорит М.Хаммер, БПР применим к “любой компании, если все ее сотрудники не умещаются за столом среднего размера”.

Уважаемый читатель, книга, которую Вы держите в руках, - первое солидное издание по обсуждаемой проблеме, выпущенное в нашей стране. До этого в России не было выпущено ни одной книги по БПР. Даже статей по БПР на русском языке - не более десятка, и излагают они только общие идеи. В связи с отсутствием книг по БПР на русском языке данное издание крайне своевременно.

Однако значение этой книги значительно больше. В ряде важных вопросов она не имеет precedентов и в мировой литературе. Большинство монографий, изданных на английском языке, посвящено описанию принципов БПР и опыту успехов и неудач, но не тому, *как* и с помощью *чего* осуществлять БПР. Единственной монографией, которая предлагала читателю не только описание идей, но и давала формализованный метод проведения БПР, является книга И. Якобсона, М. Эриксона, А. Якобсона “Объектно-ориентированный бизнес-процесс реинжиниринг” (I.Jacobson, M.Ericsson, A.Jacobson.: The Object Advantage. Business Process Reengineering with Object Technology). Однако и в ней недостаточно внимания уделено методологии и полностью отсутствует классификация и анализ инструментальных средств, применяемых при проведении БПР, описание опыта использования инструментальных средств при проведении конкретных проектов по БПР.

В предлагаемой Вам книге впервые в мировой литературе охвачен весь спектр проблем БПР: принципы, методология, формализованные модели и методы, классификация и анализ инструментальных средств, опыт их использования в конкретных проектах по БПР.

В книге описан современный подход к проведению БПР, позволяющий менеджеру принимать непосредственное участие в разработке модели компании. Суть этого подхода, названного авторами **интегрированным**, состоит в том, что для разработки моделей компании (существующей и будущей) предлагается использовать не CASE-средства, ориентированные на программиста, а интегрированные инструментальные средства, объединяющие на базе технологии динамических экспертных систем объектно-ориентированный

подход, CASE-средства, средства имитационного моделирования и “активную” графику. Именно использование в интегрированных инструментальных средствах технологии динамических экспертных систем дает возможность:

- ❶ разрабатывать модели компании непосредственно менеджерам;
- ❷ создавать динамические модели, адекватно описывающие деятельность компаний;
- ❸ автоматизировать основные этапы БПР - разработку образа будущей компании и спецификацию целей, разработку модели существующего бизнеса, разработку модели нового бизнеса, создание информационной системы поддержки; внедрение нового бизнеса.

Данная книга окажет существенную помощь менеджерам, реорганизуя бизнес, разработчикам информационных систем, поддерживающих бизнес, а также весьма полезна преподавателям, аспирантам и студентам вузов соответствующих специальностей.

*Президент Российской Ассоциации
искусственного интеллекта,
доктор физико-математических наук*

Г.С. ОСИПОВ

Семейство программных продуктов фирмы Gensym Corp. (США)



Gensym
Intelligent Real-Time Systems

• Если Вам приходится принимать оперативные решения, основываясь на информации, поступающей одновременно из сотен и тысяч источников,

• если Вы не можете ждать годы, необходимые для реализации адекватной по сложности системы управления,

• если Вы хотите эффективно использовать бесценный опыт своих самых квалифицированных специалистов,

• если Вам необходимо моделировать или реорганизовывать деятельность компании,

Вам, как и 25 компаниям, возглавляющим список крупнейших компаний мира, необходимы программные продукты фирмы Gensym (G2, GDA, ReThink, NeurOn-Line, Fault Expert), мирового лидера в области создания инструментальных средств для интеллектуальных систем управления и моделирования в реальном времени.

G2 - базовый продукт фирмы Gensym. Графическая, объектно-ориентированная среда для построения и сопровождения экспертных систем реального времени, предназначенных для мониторинга, диагностики, оптимизации, планирования и управления динамическими процессами.

Проблемно-ориентированные среды и графические языки на базе G2

G2 Diagnostic Assistant (GDA) - среда визуального программирования на базе G2 для создания приложений активной диагностики производственных и информационных процессов.

ReThink - среда моделирования и реинжиниринга деятельности компаний.

NeurOn-Line (NOL) - среда, интегрирующая технологию нейронных сетей и экспертных систем.

Fault Expert - среда создания интеллектуальных приложений, осуществляющих мониторинг и управление программным и техническим обеспечением территориально-распределенных вычислительных сетей.

АО Argussoft Co - представитель Gensym Corp. в России

Мы предлагаем полный набор услуг:

- специализированное обучение
- исследование прикладной области
- создание прототипов и приложений
- управление проектами

129090, Россия, Москва, ул. Щепкина, д. 22, подъезд 3, оф. 42

Тел.: (095) 284-8229 / 288-2436 Факс: (095) 288-2085

E-mail: g2group@arguss.msk.su





ПРЕДИСЛОВИЕ

Проектирование и управление компанией (предприятием), как и бизнес вообще, является очень сложной деятельностью, и ранее она осуществлялась интуитивно. Бизнес-процесс реинжиниринг (БПР) ставит перед собой революционную цель - превратить искусство проектирования и управления компанией в инженерную дисциплину. Для этого необходимо построить адекватные, понятные модели компании (предприятия) и дать возможность именно менеджерам, а не системным программистам, анализировать последствия изменения этих моделей. БПР обозначает совокупность методов и средств, предназначенных для кардинального улучшения основных показателей деятельности компании (предприятия) путем моделирования, анализа и перепроектирования существующих бизнес-процессов.

БПР предполагает всесторонний анализ процессов существующего бизнеса, поиск ответов на следующие основные вопросы: почему делается именно *это*? почему это делается именно *таким образом*? *что* компания хочет изменить и *почему*? и т.д.

Следует подчеркнуть, что явление, называемое БПР, *не придумано* учеными, а выявлено из практики преуспевающих компаний и формализовано учеными. По словам М.Хаммера и Дж.Чампи, первоначально БПР осуществлялся руководством преуспевающих компаний интуитивно. В этом случае риск неудач достигал 50-70 %. Для снижения риска неудач необходима разработка методологии проведения БПР, формализация методов и разработка инструментальных средств (ИС), позволяющих описать адекватную исполняемую модель компании и проанализировать последствия ее модификаций.

За очень короткое время (1990 - 1996 гг.) БПР стал одной из самых популярных тем на конференциях по теории менеджмента и по разработке информационных систем. В этот период за рубежом появилось более десятка книг и сотни статей. Однако большинство из них посвящено изложению общих идей БПР и описанию опыта успехов и неудач. Мало изданий посвящено тому, как проводить БПР, как и какие модели компании следует разрабатывать, какие работы выполнять на каждом из этапов БПР. И совсем отсутствуют работы, в которых бы рассматривалась методология БПР, приводились классификация, анализ и описание ИС, ориентированных на проведение

БПР; освещались вопросы использования ИС при проведении реинжиниринга компаний.

Данная книга отличается от других книг по БПР, изданных за рубежом, именно тем, что в ней решаются перечисленные выше проблемы.

В связи с тем что это первая книга по БПР на русском языке, в ней последовательно изложены все основные вопросы БПР: идеи, принципы, методология, формализованные методы, анализ инструментальных средств, опыт выполнения конкретных проектов.

Особое внимание уделяется правильному выбору ИС, что объясняется следующими обстоятельствами:

- Проведение БПР без развитых ИС увеличивает риск неудачи до 50-70 %;

- Многие ИС автоматизируют только отдельные этапы БПР, что также увеличивает риск неудачи и время проведения БПР;

- Большинство ИС ориентированы на программистов и не предполагают непосредственное участие менеджеров в разработке моделей компании, что зачастую приводит к неадекватности моделей и к непоправимым ошибкам при проведении БПР;

- Далеко не все ИС имеют выразительные средства, достаточные для того, чтобы адекватно описать деятельность конкретной компании. Например, практически все CASE-средства, используемые при разработке моделей компаний, не имеют выразительных возможностей, позволяющих описать и промоделировать (если это необходимо) динамику процессов;

Книга ориентирована на три группы читателей:

① На менеджеров верхнего звена, которые хотят получить ответы на вопросы типа: *что такое БПР? зачем и кому он нужен? что он дает? в чем суть БПР? каковы факторы успеха и риска при проведении БПР? каковы последствия БПР для компании? какую роль играют информационные технологии при проведении БПР? кто осуществляет БПР?* Ответы на эти вопросы даны в главах 1 и 2.

Глава 1 вводит читателя в круг проблем БПР. Здесь приведены определения, примеры успехов и неудач БПР, показано, чем БПР отличается от предыдущих подходов к реорганизации компаний. В главе 2 представлено соотношение инжиниринга, реинжиниринга и усовершенствования. Показана решающая роль информационных технологий в успешном проведении БПР. Определена структура традиционной и перепроектированной компании. Рассмотрены особенности БПР и его влияние на структуру традиционной компании.

② На менеджеров, которые хотят получить более детальное представление о БПР, достаточное для того, чтобы руководить проектом по БПР. Этой категории читателей следует изучить кроме двух первых глав главы с 3-й по 5-ю и приложение 2.

В главе 3 обоснована необходимость моделирования бизнеса при проведении БПР. Определены требования к модели компании со стороны различных участников. Рассмотрена ограниченность традиционных подходов к созданию моделей компании на основе использования CASE-средств. Указаны достоинства и ограничения объектно-ориентированного подхода к БПР. Показано, что для адекватного представления моделей компании необходимо выполнение следующих условий:

- средства должны быть ориентированы не столько на программиста, сколько на менеджера;
- средства должны интегрировать на базе технологии экспертных систем объектно-ориентированный подход, CASE-средства, средства имитационного моделирования, средства активной (анимационной) графики.

В главах 4 и 5 описана структура БПР и его основные этапы, включающие: спецификацию целей и разработку образа будущей компании, разработку модели существующего бизнеса, разработку модели нового бизнеса, внедрение нового бизнеса. Определены два вида моделей компании - внешняя и внутренняя, а также те сущности, которые необходимо уметь представлять в моделях компании.

В приложении 2 на примере компании АО "Шереметьево-Карго" описывается модель, предназначенная для проведения комплексных аналитических исследований эффективности функционирования подразделений и служб типовой транспортной компании. Модель разработана АО "Аргуссофт Компани" на базе ИС ReThink (фирма Gensym, США).

③ На разработчиков проектов БПР и разработчиков информационных систем компании, которым следует прочитать, кроме глав 1 - 5 и приложения 2, главы 6 - 9 и приложение 1.

Глава 6 описывает обратный инжиниринг, состоящий в разработке моделей (внешней и внутренней) существующей компании, а глава 7 - прямой реинжиниринг, заключающийся в разработке модели вновь проектируемой компании. Учитывая важность для БПР информационных технологий, вопросу разработки информационной системы поддержки бизнеса посвящена глава 8.

В главе 9 представлены основные методологии, используемые при проведении БПР. Указано, что нет единого подхода, пригодного для всех приложений. Определена роль ИС, проведена их классификация и показано, что далеко не все ИС автоматизируют основные этапы БПР. На примере ИС ReThink подробно излагается интегрированный подход к БПР, обеспечивающий автоматизацию всех основных этапов БПР и разработку адекватных моделей компании менеджерами, а не программистами. ReThink позволяет менеджеру описывать модель компании на основе заранее определенных базовых компонентов.

Приложение 1 посвящено описанию системы G2 (фирма Gensym - США), на основе которой разработано ИС ReThink. Система G2 служит средством для создания интеллектуальных приложений в управлении и бизнесе. Если имеющихся в ReThink компонентов недостаточно для описания модели, то они могут быть определены (программистами, а не менеджерами) средствами G2.

Издание данной книги стало возможным благодаря гранту 96-01-14068, полученному в Российском фонде фундаментальных исследований в 1996 г.

Авторы считают своим приятным долгом выразить благодарность президенту фирмы Gensym (США) Р.Муру (R. Moore) за предоставленные материалы по БПР и генеральному директору АО "Шереметьево-Карго" В.И.Новикову - за разрешение опубликовать в данной книге материалы совместных с АО "Аргуссофт Компани" работ по моделированию компании "Шереметьево-Карго".

Хотя оба автора несут полную ответственность за все написанное в данной монографии, тем не менее считают нужным указать, что Е.Г.Ойхманом определена общая структура книги и написаны предисловие и главы 3, 5, 7, а Э.В.Поповым - главы 1, 2, 4, 6, 8, 9 и приложение 1. Автор приложения 2 - Е.Б.Кисель, сотрудник АО "Аргуссофт Компани".

Е.Г.ОЙХМАН
Э.В.ПОПОВ

ВВЕДЕНИЕ В ПРОБЛЕМУ

1.1

Реинжиниринг бизнеса - новое направление теории менеджмента

Современные компании (предприятия, организации, фирмы, корпорации) в значительной мере все еще базируются на принципах, вытекающих из теории Адама Смита. Напомним, что более 200 лет тому назад этот экономист создал теорию для систематического описания стихийно сложившихся способов организации производства, которые возникли за несколько столетий до него. В своем фундаментальном труде “Благосостояние нации”, опубликованном в 1776 г., А.Смит сформулировал принципы организации труда в промышленности, которые оказались по-настоящему революционными для того времени. Производственный процесс предлагалось разбить на элементарные, простые задания (работы), чтобы каждое из них мог выполнять один рабочий. От рабочего не требовалось высокой квалификации и умения выполнять работу в целом - достаточно, чтобы он специализировался на одном или на нескольких простейших заданиях. Это легко реализуемая идея, в связи с чем предложенные принципы были и остаются весьма эффективными в массовом производстве типовой продукции, выполняемой силами большой армии низкоквалифицированных рабочих, использующих простое оборудование.

Однако принципы, заложенные А. Смитом, не соответствуют требованиям современной индустрии, так как продукция в наше время перестает быть массовой и должна ориентироваться на узкие группы потребителей; исполнители хорошо образованны, стремятся к ответственности и решению по-настоящему сложных задач; рынок продуктов стал намного шире, а конкуренция и борьба за потребителя - более агрессивной.

Мир, в котором живут современные предприниматели, даже за последние 10 лет существенно изменился в следующих аспектах.

① **Клиенты в цивилизованном мире взяли в свои руки контроль на рынке.** Они намного лучше, чем в начале 80-х годов, осведомлены о своем положении на рынке и о тех возможностях выбора продукции, которые у них имеются. Они больше не допускают того, чтобы их рассматривали как часть безликой массы; они ожидают, что с ними будут обращаться как с индивидуальностями. Как сказал М.Хаммер [5], “больше не осталось понятия “клиент вообще”, теперь есть только “именно этот клиент””.

② **У клиентов сформировались новые ожидания относительно предлагаемых им товаров и услуг.** Каждый отдельный клиент нуждается в продукции, которая:

- адаптирована и сконфигурирована таким образом, чтобы удовлетворять определенные потребности данного клиента;

- поставляется способом, наиболее подходящим для клиента;

- поставляется только тогда, когда клиент хочет ее получить.

③ **В современном мировом рынке конкуренция присутствует буквально повсюду.** Товары, интересующие клиентов, уже перестали быть локальными и производятся по всему миру. Фирма не может уступать ни в чем своим конкурентам, независимо от того, где они находятся.

Наиболее существенными оказались изменения, затронувшие средства производства и технологии, а среди последних прежде всего информационные технологии. **Информационные технологии** (ИТ) - это не просто база многих других важных технологий (персональных и портативных компьютеров, сетей и средств коммуникации, робототехники, распределенных баз данных и т.д.), но и способ, с помощью которого информация предлагается клиентам. Информация стала (или станет в ближайшем будущем) важной составляющей частью товаров и услуг, предоставляемых компаниями на рынок.

Многие фирмы продолжают по инерции держаться за старые идеи в практике управления. Однако чтобы выжить в новом мире, фирмы должны кардинально переосмыслить способы организации своего бизнеса. Настало время, когда необходимо похоронить старые способы мышления и заменить их принципиально новым подходом, который позволит в полной мере реализовать преимущества новых технологий и человеческих ресурсов. Этот подход - стержень и основа инжиниринга (англ. engineering) бизнеса. В данной главе рассматриваются фундаментальные понятия **инжиниринга бизнеса** и его наиболее важное применение - реинжиниринг, или перестройка, существующих компаний; обсуждаются факторы успеха и риска в их деятельности. В деловом мире реинжиниринг бизнеса, а точнее, **реинжиниринг бизнес-процессов** - БПР, стал “притчей во языцех” 90-х гг., так как он оказывает огромное воздействие на любой бизнес. Каж-

дый говорит о нем, каждый имеет свою собственную интерпретацию и каждый готов его осуществить. На самом же деле далеко не всем удалось достичь успеха при проведении БПР на практике.

По своим потенциальным возможностям реинжиниринг так же революционен, как и концепция Just-IN-Time Manufacturing, основные принципы которой были разработаны японской компанией Тойота в конце 70-х гг. Оба подхода направлены на замену функционального тейлоризма идеями синхронизации производства с потребностями потребителя.

С массовым распространением концепций менеджмента, ориентированных на процесс - Just-IN-Time Manufacturing (производство "как раз вовремя"), Total Quality Management (глобальное управление качеством) и Business Process Reengineering (реинжиниринг бизнес-процессов) - руководство компаний осознало, что деятельность составляет единый процесс, улучшая составные элементы которого можно достичь преимуществ в конкуренции.

1.1.1

Понятия "инжиниринг бизнеса" и "реинжиниринг бизнеса"

Инжиниринг бизнеса - это набор приемов и методов, которые компания использует для проектирования бизнеса в соответствии со своими целями. Необходимость проведения инжиниринга можно пояснить следующим образом. Наиболее фундаментальная движущая сила каждой компании - потребность улучшения своего финансового положения. Говоря проще, компания должна делать деньги.

Компания зарабатывает больше денег, если выпускает свою продукцию настолько эффективно, насколько это возможно. В конечном счете это означает, что нужно снижать стоимость своих бизнес-процессов, сокращать избыточные внутрифирменные работы, делать труд работников более производительным путем улучшения их профессиональной подготовки и повышения ответственности. Компания повышает свой доход за счет увеличения объема продаваемых товаров и (или) услуг.

Чтобы выжить в современных условиях, фирма должна адаптироваться и постоянно приспосабливаться к изменяющемуся окружению. Для этого недостаточно просто выдавать руководящие указания с вершины многоуровневой иерархии компании. Здесь нужен инжиниринг бизнеса. Компания должна быть организована таким образом, чтобы отслеживать постоянные изменения во внешнем мире.

Компания не может быть стабильной, она должна изменяться постоянно, чтобы удовлетворять запросы потребителей, не уступать

сопернику в условиях жесткой конкуренции, усовершенствовать свои внутренние процессы, развивать диапазон предлагаемых товаров и услуг и ставить перед персоналом реальные цели, предоставляя для их достижения творческую свободу действий в рамках некоторого “остова”, обеспечивающего фокусирование внимания на тех бизнес-процессах, которые ориентированы на клиентов.

Инжиниринг бизнеса представляет собой множество методик, используемых для проектирования бизнеса, удовлетворяющего заданным целям компании. Эти методики включают:

- ① пошаговые процедуры для проектирования бизнеса;
- ② систему обозначений (язык), описывающую проектирование бизнеса;
- ③ эвристики и прагматические решения, позволяющие измерить степень соответствия спроектированного бизнеса заданным целям.

Итак, инжиниринг бизнеса направлен на организацию коммерческого предпринимательства на конкурентоспособной основе. На первый взгляд здесь нет ничего нового. Предприниматели всегда стремились к конкурентоспособности. Однако только **реинжиниринг предусматривает новый способ мышления** - взгляд на построение компании как на инженерную деятельность. Компания рассматривается как нечто, что может быть построено, спроектировано или перепроектировано в соответствии с инженерными принципами. Сама мысль о том, что можно более эффективно участвовать в конкурентной борьбе, если при проектировании своей компании будут использоваться принципы современной инженерии - принципы, основанные на четко организованных процессах, - является революционной. Ниже будет показано, что риск при проведении инжиниринга достаточно велик, но те улучшения, которых можно добиться с помощью новых способов мышления, оказываются кардинальными. Под “кардинальными” понимают улучшение на порядок - по крайней мере в 10 раз; это не менее чем 90%-ное сокращение стоимостных или временных затрат либо 90%-ное повышение качества, а не 10%-ное улучшение этих показателей.

Объектом реинжиниринга являются процессы, а не компания. Компания проводит реинжиниринг не отделений производства или продажи, компания осуществляет реинжиниринг работы, которую выполняют люди в этих подразделениях.

М.Хаммер предложил такое определение [5]: “реинжиниринг - это фундаментальное переосмысление и радикальное перепроектирование деловых процессов для достижения резких, скачкообразных улучшений в решающих, современных показателях деятельности компании, таких, как стоимость, качество, сервис и темпы”. Это определение содержит четыре ключевых слова: фундаментальный, радикальный, резкий (скачкообразный) и процесс (наиболее важное слово).

Ключевое слово: фундаментальный

На начальной фазе реинжиниринга необходимо ответить на следующие фундаментальные вопросы о деятельности компании:

- ❶ Почему компания делает то, что она делает?
- ❷ Почему компания делает это (то, что она делает) таким способом?
- ❸ Как ей хочет стать компания?

Отвечая на эти вопросы, специалисты должны выявить и переосмыслить правила и предположения (зачастую явно не выраженные), положенные в основу текущего способа ведения бизнеса. Часто эти правила оказываются устаревшими, ошибочными или неуместными.

Реинжиниринг ничего не принимает на веру. Реинжиниринг не начинается с предположений или чего-нибудь заданного. Проводя реинжиниринг, надо так формулировать задачи, чтобы в них неявно не содержалось предположений о процессе. Так, например, задача “Как более эффективно выполнить проверку пользовательского кредита?” содержит предположение, что надо осуществлять проверку кредита. Хотя во многих случаях цена проверки может превосходить потери, являющиеся следствием отсутствия проверки. Итак, реинжиниринг начинается с того, *что* компания делает, *как* она это делает и затем концентрируется на том, *что должно быть*.

Ключевое слово: радикальный

Радикальное перепроектирование означает перепроектирование, затрагивающее суть (корень) явлений, а не поверхностные изменения, т.е. в ходе радикального перепроектирования отбрасываются все существующие структуры и процедуры и предлагается совершенно новый способ выполнения работы. Итак, реинжиниринг - это изобретение, а не улучшение, увеличение или модификация.

Ключевое слово: резкий (скачкообразный)

Реинжиниринг не применяется в тех случаях, когда необходимо получить улучшение либо увеличение некоторых показателей деятельности компании на 10 - 100 %. При этом используются более традиционные методы (например, “управление качеством”), применение которых не сопряжено со значительным риском.

Проведение реинжиниринга целесообразно только в тех случаях, когда требуется достичь резкого (скачкообразного) улучшения показателей деятельности компании путем замены старых методов управления на новые.

Можно выделить три типа компаний, для которых применение реинжиниринга необходимо и целесообразно.

- ❶ Компании, находящиеся на грани краха в связи с тем, что цены на товары заметно выше, чем у конкурентов и (или) качество товаров (сервис) заметно ниже, чем у конкурентов. У этих компаний нет выбора: если они не предпримут решительных шагов, они неизбежно

разорятся. В подобном положении находилась в начале 80-х гг. компания Ford Motor. Успешный опыт этой компании по проведению реинжиниринга будет описан ниже (см. разд. 1.2).

② Компании, не находящиеся в текущий момент в затруднительном положении, но руководство компаний предвидит неизбежность возникновения трудноразрешимых проблем, связанных, например, с появлением новых конкурентов, изменением требований клиентов, изменением экономического окружения и т. п. Примером такой компании была в конце 80-х гг. компания Aetna Life & Casualty [5].

③ Компании, не имеющие проблем ни сейчас, ни в ближайшем обозримом будущем. Это компании-лидеры, проводящие агрессивную политику. Они не удовлетворяются текущим хорошим состоянием и с помощью реинжиниринга хотят добиться лучшего.

Ключевое слово: процесс

В связи с тем что это слово является наиболее важным в определении понятия “реинжиниринг”, посвятим ему отдельный подраздел.

1.1.2

Понятие “бизнес-процесс”

Хотя понятие “процесс” (“бизнес-процесс”) является в данном контексте наиболее важным, его трудно объяснить менеджерам, так как большинство из них привыкло иметь дело с задачами, работами, структурами, людьми, но не с процессами.

Бизнес-процесс - это множество внутренних шагов (видов) деятельности, начинающихся с одного или более входов и заканчивающихся созданием продукции, необходимой клиенту. Назначение каждого бизнес-процесса состоит в том, чтобы предложить клиенту товар или услугу, т.е. продукцию, удовлетворяющую его по стоимости, долговечности, сервису и качеству. Термин *клиент* следует понимать в широком смысле. Это может быть действительно просто клиент, а может быть и другой процесс, протекающий во внешнем окружении компании, например у партнеров или субподрядчиков.

В понятии процесса нет ничего нового. Каждая компания всегда имела свои процессы. Проблема состоит в том, что процессы не удастся описывать так же легко, как организационные иерархические структуры. У организационных подразделений есть “имена” (“производство продукции”, “доставка продукции”), с ними связаны ответственные должностные лица (“президент”, “начальник подразделения”). Процессы же обычно невидимы, не имеют описаний и имен. Однако понятие “процесс” возникает более естественным образом, чем организационные иерархии тогда, когда люди коопери-

руются для достижения обещанного клиенту результата. При традиционной структуре внимание фокусируется на заданиях, работах, людях, на структурах, но не на процессах, хотя процессы пронизывают традиционные организационные структуры (рис. 1.1).

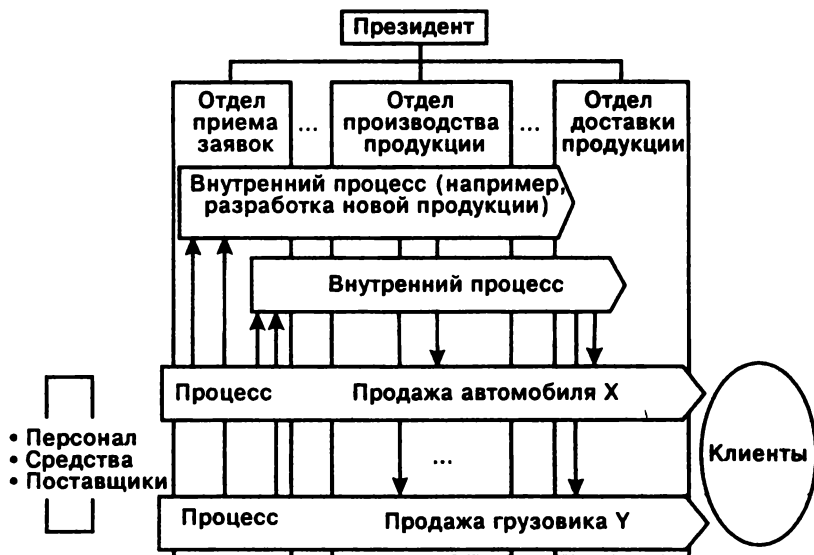


Рис.1.1. Процессы и традиционная структура компании

Т.Давенпорт [3] выразил эту мысль так: “В то время как иерархическая структура организации обычно представляет собой временной срез распределения ответственности и взаимоотношений по отчетности, структура ее процессов отражает динамический взгляд на то, как эта организация производит продукцию”. Как отмечает Т.Давенпорт, “*процесс* - это специфически упорядоченная совокупность работ, заданий (activities) во времени и в пространстве, с указанием начала и конца и точным определением входов и выходов”. К этому определению можно добавить, что *выходом* может быть либо товар, либо услуга. Новый автомобиль является примером товара, а ремонт автомобиля - услуги. Входы и выходы процесса могут взаимодействовать как с конкретным клиентом, так и с некоторым другим процессом во внешнем окружении компании - но не с другим внутренним процессом.

Чтобы проще рассуждать о процессах компании, им можно дать описательные имена-характеристики. Например, М.Хаммер и Дж.Чампи [5] предлагают именовать процессы в соответствии с их начальным и конечным статусом, например: “*разработка продукта: от требований на продукт - к продукту*” или “*продажа: от заявки - к*

заказу". Такие имена позволяют отделить процессы от подразделений. Позже, когда мышление в терминах процессов станет привычным, можно будет обозначать их более простыми именами. Для именования процессов целесообразно использовать отглагольные формы: чтобы отличать их от имен подразделений.

Инжиниринг бизнеса, таким образом, требует отойти от старой организации работы, предложенной А. Смитом. В бизнес-процессы необходимо объединить связанные задания. Однако сборка единого процесса из отдельных его фрагментов (заданий) - отнюдь не простая задача: часто не ясно, как это следует делать.

В компаниях с традиционной структурой руководители подразделений - как правило, мужчины - ведут явно или неявно борьбу за то, чтобы иметь больше денег, больший штат и больше власти. А мужчин, как известно, с незапамятных времен с детства приучали к соперничеству, в то время как женщин - к сотрудничеству. Наверное, в этом и заключается причина того, что очень мало женщин занимают руководящие посты в современных компаниях. Некоторые авторы, включая Дж.Карлсона - одного из создателей теории инжиниринга бизнеса, воплотившего ее идеи в полной перестройке Скандинавских Авиалиний (SAS), - отмечают, что компании, ориентированные на процессы, в большей степени соответствуют стилю работы, характерному для женщин: "Новая роль руководителей в подобных компаниях должна открыть новые возможности в бизнесе для женщин" [2].

По сравнению с традиционным подходом процессы обладают и другими преимуществами. Действительно, трудно или невозможно измерить достоинства иерархической структуры или улучшить ее, в то время как при ориентации на процессы специалисты имеют дело с такими четко оцениваемыми характеристиками, как *стоимость, длительность, выход, качество и степень удовлетворения клиента*.

Другое интересное определение процесса связано с понятием "потока ценностей" (value stream), введенным Дж. Мартином [9]. *Поток ценностей* - это "множество законченных (end-to-end) состыкованных действий (action), которые в совокупности создают некоторую продукцию, имеющую потребительскую ценность для клиента". Дж.Мартин отказался от термина "процесс", поскольку последний, по его мнению, имеет слишком много различных значений, что может приводить к недоразумениям.

И.Якобсон [7] предлагает делить процессы на *внешние*, называемые им *прецедентами*, и *внутренние*. В дальнейшем изложении будем придерживаться этой терминологии.

Процесс, ориентированный на клиента, выражается в терминах, отражающих потребности индивидуального, а не массового клиента. Концентрация на процессах, производящих ценности (продукцию) для клиентов, а не на каких-либо других частях биз-

неса, позволяет наилучшим образом выполнить то, что должно быть сделано фирмой в конечном итоге. Приняв этот способ фокусирования внимания, можно увидеть, что значительная часть работы, выполняющаяся в рамках старой компании, не связана с созданием потребительских ценностей и решает некоторые внутренние проблемы компании.

Дж.Карлсон [2], бывший президент SAS, организовал свой персонал в команды, по одной на каждый авиарейс. Команда несла ответственность за процесс "проведение полета: от регистрации до выдачи багажа по прибытии". Команда состояла из персонала, работавшего как в самолете, так и на земле. К последним относились, например, техники и сотрудники по приему и выдаче багажа. Это был принципиально новый подход к организации коммерческих рейсов по сравнению с процедурами в традиционных компаниях. Дж.Карлсон довольно ярко описывает, как небольшая неисправность, например, недостаточно закрученная деталь, отрабатывается в рамках традиционной и новой компании. Ранее ответственный за техническую часть должен был представлять формальный рапорт о неисправности, в результате чего запускалась длительная и дорогостоящая процедура, в которую вовлекались менеджеры на нескольких уровнях иерархии. В новой компании ответственному достаточно просто попросить техника подкрутить деталь, сняв тем самым все проблемы.

Рассмотрим теперь, как соотносятся процессы с товарами и услугами. Товары - п р о д у к т ы компании - проектируются, производятся, продаются и поставляются; а п р о ц е с с ы занимаются их обработкой. В западном мире традиционно обращают внимание на товары (результат), а не на процессы. Можно сказать, что в традиционной компании процессы формируют под спроектированные товары. В настоящее время осознана нецелесообразность подобного подхода: товар и процесс должны строиться совместно и согласованно друг с другом. Поясним эту мысль на следующем примере [7].

В начале 50-х гг. IKEA, тогда еще небольшая шведская компания по розничной продаже мебели, выработала уникальную концепцию бизнеса. В то время хорошая мебель продавалась в шикарных магазинах, расположенных в центре города. Объем продаж был небольшим, поэтому цены неизбежно оказывались довольно высокими. Ингвар Кампрад, основатель IKEA, хотел продавать свою мебель всем покупателям, независимо от их доходов и места жительства. Чтобы снизить цены, И.Кампрад должен был увеличить объем продаж и снизить издержки. Он захотел избавиться от посредников и продавать мебель непосредственно клиенту, который мог взять на себя доставку и установку мебели. Решение оказалось простым и гениальным. Мебель производилась и упаковывалась таким образом, чтобы для ее хранения требовалось минимальное пространство на складе, чтобы ее было дешевле перевозить и проще собирать. Клиенты должны были сами забирать мебель либо со склада IKEA, либо с местного железнодорожного вокзала, перевозить ее к себе домой и там собирать. Используя этот метод и организовав продажу через каталоги, отправляемые по почте, а не через

дорогие магазины, И.Кампрад сумел существенно снизить себестоимость. Он обнаружил, что может целиком изменить процесс продажи и распределения, кардинально увеличив при этом конкурентоспособность своей компании. Для реализации этих идей необходимо было выполнить, по крайней мере, два условия. Во-первых, что самое главное, клиенты должны были принять идею самостоятельной доставки и сборки мебели. Во-вторых, следовало приспособить конструкцию мебели к новому процессу распределения: конструкция мебели должна была обеспечить простоту сборки и компактность упаковки. И.Кампрад считал все это осуществимым, и время доказало его правоту. Он изменил способ того, как люди меблируют свои дома. Сейчас IKEA - международная компания, оборот которой в 1993 г. составил 4 млрд. долл. США.

Очевидно, что товары компании должны проектироваться в согласии с процессами этой компании. Наиболее важными процессами (прецедентами) являются те, которые связаны непосредственно с клиентами.

Процессы позволяют компаниям предлагать услуги - точно так же, как процессы производят товары. В этом смысле услуги не отличаются от товаров. Традиционно товары рассматриваются как осязаемые результаты деятельности компании, а услуги - как нечто неосязаемое. Товары оплачиваются индивидуально, а услуги - за единицу рабочего времени (дни, часы). Однако подобный способ мышления оставляет большую зону неопределенности, в пределах которой услуги могут рассматриваться как товары (авиапутешествия, например). Кроме того, производящие компании начинают усматривать финансовую выгоду в сопровождении своих товаров определенной совокупностью услуг.

Таким образом, черта, разделяющая товары и услуги, оказывается иллюзорной. В контексте инжиниринга бизнеса удобнее рассматривать товары (продукты) и услуги как два различных вида продукции. Под *продукцией* понимается результат бизнеса, имеющий ценность, материальную или нематериальную, для некоторого конкретного клиента. Согласно этому определению все то, что привыкли считать услугами, можно рассматривать как частный вид продукции.

1.2

Примеры успешного реинжиниринга

Ниже описаны примеры успешного реинжиниринга, которые проиллюстрируют как он работает и что он может дать компании. При рассмотрении каждого из этих примеров рекомендуем читателям помнить и выделять четыре ключевых слова: “фундаментальный”, “радикальный”, “резкий/скачкообразный” и

особенно “процесс”. Опишем опыт применения реинжиниринга в следующих компаниях: IBM Credit, Ford Motor, Kodak [5].

Опыт IBM Credit. IBM Credit Corporation является филиалом IBM и занимается кредитованием клиентов, которым IBM продает компьютеры, программы и предоставляет услуги. Это весьма доходный бизнес. Проблема IBM Credit состояла в том, что при существующем технологическом цикле решение вопроса о кредитовании клиента занимало в среднем 6 дней, а в сложных случаях - до двух недель. Чрезмерная длительность принятия решения приводила к потере клиента, так как он за это время мог найти (и часто находил) другой источник финансирования. Кроме того, компания при существующем технологическом цикле не могла ответить на вопрос клиента, на каком шаге обработки находится его запрос и когда будет дан ответ?

Длительность принятия решения по запросу клиента была вызвана тем, что обработка запроса осуществлялась в пять шагов, выполняемых последовательно в пяти различных подразделениях компании. При этом передача запроса из одного подразделения в другое осуществлялась на бумажном носителе. Сначала компания предприняла попытку отследить текущее состояние запроса, для чего по завершении каждого шага информация об этом посылалась администратору. Это решило одну проблему - “На каком шаге обработки находится запрос?”, но увеличило время обработки запроса до семи дней.

После мучительных раздумий два старших менеджера IBM Credit решили сами пройти с несколькими запросами клиентов все пять шагов. При этом они просили исполнителей обрабатывать запросы без задержки. Эксперимент показал, что собственно на обработку запроса затрачивается всего 90 мин., а остальное время расходуется на передачу запроса из одного подразделения в другое. Таким образом, оказалось, что проблема заключена не в сути решаемой задачи и не в эффективности, с которой работают исполнители, а в структуре процесса обработки. Итак, для решения проблемы необходимо было изменить процесс, а не его отдельные шаги.

Проведенный анализ позволил выявить предположение (явно не выраженное), лежащее в основе используемого способа обработки: каждый запрос является сложной задачей, требующей для ее решения участия экспертов разных специальностей. Анализ запросов показал, что это предположение ошибочно, так как большинство запросов являются простыми и их обработка сводится к работе с базой данных, что может сделать клерк, а не эксперт. Таким образом, существующий процесс обработки был ориентирован на наиболее трудный запрос. В новом процессе всю обработку выполняет один специалист, снабженный информационной экспертной системой, обеспечивающей принятие решения и доступ ко всем необходимым данным и инструментариям. Теперь в большинстве случаев (более 90% запросов

сов) один специалист обеспечивает решение задачи. В трудных случаях специалист обращается к экспертам.

Итак, в результате реинжиниринга IBM Credit радикально перепроектировала процесс обработки и достигла скачкообразного улучшения основных показателей деятельности компании: время обработки запроса сокращено с семи дней до четырех часов, количество обрабатываемых запросов возросло в сто раз (при небольшом уменьшении количества сотрудников).

Опыт Ford Motor. В начале 80-х гг. компания Ford подобно многим другим компаниям Америки искала способы сокращения административных расходов. Компания Ford верила, что расходы можно сократить в отделении оплаты счетов (account payable department). На тот момент в североамериканском отделении оплаты счетов работало более 500 человек. Руководство Ford предполагало, что использование компьютеров позволит сократить число сотрудников на 20 %. По приведенному выше определению, это не реинжиниринг, а наращиваемое улучшение, автоматизирующее существующий ручной процесс.

Руководство Ford до посещения компании Mazda считало, что 20 %-ное сокращение численности работающих - вполне хороший результат. Но посетив эту компанию, руководители Ford отметили, что там оплатой счетов занимается всего 5 человек (вместо 500 человек в компании Ford), хотя Mazda меньше, чем Ford, явно не в 100 раз. В связи с этим руководство Ford приняло решение переосмыслить весь процесс, в котором участвует отделение оплаты счетов.

Итак, начав с автоматизации работы отделения, компания Ford пришла к выводу о необходимости проведения реинжиниринга бизнес-процессов. Отделение оплаты счетов не могло быть подвергнуто реинжинирингу, так как это подразделение, а не процесс. Процесс, в котором участвует это отделение, называется "*поставки*". Этот процесс начинается с того, что департамент заказов посылает продавцу товаров заказ на их приобретение. При этом копия заказа направляется в отделение оплаты счетов. Когда продавец отправил товары и они прибыли в компанию Ford, клерк из отдела получения товаров составляет документ получения, описывающий товары, и отправляет его в департамент оплаты счетов. Тем временем продавец посылает в отделение оплаты счетов накладную на товары (invoice).

К этому времени в отделении оплаты счетов находится три документа на эти товары: заказ на приобретение, документ получения и накладная. Если все три документа соответствуют друг другу, то клерк оплачивает счет. В большинстве случаев именно эта ситуация и имеет место.

При несоответствии документов необходимо найти источник ошибки. Основное время в своей работе клерк тратит на обработку ситуаций, в которых документы не соответствуют. Обработка одной такой ситуации часто занимает более недели. Здесь уместно вспом-

нить *правило 80-20*, сформулированное итальянским экономистом Vilfredo Pareto. Это правило утверждает, что *“80 % усилий затрачивается на 20 % работы.”*

Новый процесс оплаты счетов, разработанный в компании Ford в ходе реинжиниринга, радикально отличается от старого. Клерки теперь не сопоставляют заказ на приобретение с документом получения, так как в новом процессе устранена накладная. Это привело к существенному уменьшению количества сотрудников (125 человек вместо 500).

Новый процесс состоит в следующем. Отделение заказов посылает продавцу заказ на приобретение товаров и одновременно вводит этот заказ в базу данных. Затем продавец посылает заказанные товары в отдел получения. Когда товары прибывают в отдел получения, клерк через компьютерный терминал проверяет соответствие присланных товаров товарам, перечисленным в заказе и хранящимся в базе данных. Если соответствие есть, то клерк принимает товары и вводит информацию об этом в базу данных. Компьютер, получив информацию о прибытии товаров, автоматически отправляет продавцу чек об оплате товаров. Если соответствия нет, то клерк отвергает груз и отправляет его обратно продавцу.

Суть изменений, проведенных компанией Ford, довольно проста и состоит в авторизации оплаты, выполняемой в отделе получения. Фактически новый процесс подводит к устранению департамента оплаты счетов.

Процесс реинжиниринга в компании Ford отменил неявно выраженное, но неукоснительно соблюдаемое правило бизнеса: *“Мы платим, когда мы получаем накладную”*. Новое правило таково: *“Мы платим, когда мы получаем товары”*.

Необходимо подчеркнуть, что **изменения процессов, выполненные в компаниях Ford и IBM Credit, были бы невозможны без новых информационных технологий**. Роль новых информационных технологий в реинжиниринге более подробно рассмотрена в разд. 2.2.

Опыт Kodak. В 1987 г. основной конкурент Kodak компания Fuji объявила о выпуске новой 35-мм камеры. Компания Kodak не вела исследований в этом перспективном направлении. Традиционный для Kodak цикл от начала разработки нового изделия до его производства составлял 70 недель. Такое длительное отставание от Fuji позволило бы последней получить большие преимущества на новом перспективном рынке. Для того чтобы сократить этот цикл, Kodak решила провести реинжиниринг процесса разработки нового продукта.

Разработка продукта может выполняться последовательно или параллельно. При последовательной разработке весь проект разбивается на шаги и переход к очередному шагу осуществляется только тогда, когда полностью завершен предыдущий. Очевидно, что при последовательном подходе время разработки больше, чем при па-

раллельном, но при этом подходе меньше объем работ, так как не приходится устранять несоответствия между компонентами, разработанными на параллельно выполняемых шагах. Как правило, несоответствия неизбежны, и их устранение может свести на нет преимущества параллельного подхода.

Компания Kodak решила использовать при разработке нового изделия последовательно-параллельный подход, т.е. некоторые части камеры разрабатывались одновременно. Для того чтобы ускорить выпуск нового изделия, компания Kodak приняла решение о реинжиниринге процесса разработки изделия, используя технологию CAD/CAM (Computer Aided Design/Computer Aided Manufacturing). Эта технология позволяет проектировать изделия непосредственно на экране компьютера, не прибегая к чертежам на бумаге, что значительно ускоряет разработку.

Использование технологии CAD/CAM и интегрированной базы данных (БД), хранящей текущее состояние проекта, позволило компании Kodak применить при разработке ряда компонентов камеры параллельный подход. Каждый день в БД добавлялись результаты, полученные параллельно работающими группами. Каждый вечер группа проектировщиков инспектировала БД с целью поиска несоответствий между результатами работы параллельно работающих групп. Если несоответствия обнаруживались, то они тут же исправлялись. При используемой ранее технологии разработки несоответствия могли быть обнаружены только через недели или месяцы, т.е. тогда, когда параллельно разработанные части собирались вместе.

Новый процесс разработки, использованный компанией Kodak, называется *одновременной инженерией*. Этот подход использовался ранее в космической индустрии, но компания Kodak впервые применила его для разработки товаров широкого потребления. Новый процесс разработки позволил сократить срок выпуска нового продукта с 70 недель до 38 недель. Более того, так как новый процесс позволяет промоделировать сборку продукта до его изготовления, стало возможным выбирать те конструкции, которые проще и дешевле в производстве. Благодаря этому Kodak уменьшил стоимость вновь спроектированной камеры на 25 %.

Все рассмотренные выше примеры успешного бизнес-реинжиниринга соответствуют приведенному выше определению реинжиниринга и могут быть охарактеризованы четырьмя аспектами, свойственными реинжинирингу:

① *Ориентация на процесс.* Во всех трех случаях положительный результат получен не благодаря ориентации на узкую задачу, решаемую в predetermined организационных границах, а благодаря рассмотрению всего процесса в целом.

② *Амбиции.* Все компании ставили перед собой цель обеспечить качественный прорыв в решении задачи, их не устраивало улучшение на 10 - 50 %, обеспечиваемое автоматизацией.

- ③ *Отказ от устоявшихся правил.* Каждая компания при проведении реинжиниринга неизбежно приходила к необходимости отказаться от установленных ранее незыблемых правил ведения бизнеса.
- ④ *Информационные технологии.* Средствами, которые позволили каждой из рассмотренных выше компаний отказаться от устаревших правил и создать модели новых процессов, скачкообразно улучшив основные показатели, явились информационные технологии.

1.3

Условия успешного реинжиниринга и факторы риска

1.3.1

Факторы успеха

Систематических исследований об эффективности попыток реинжиниринга пока не проведено. Однако экспертные оценки показывают, что ранее (в 1993 г.) около 50 % проектов по реинжинирингу заканчивались неудачей [5]. С целью выяснения причин неудач и определения необходимых для успеха предпосылок проведен ряд специальных исследований [4], опирающихся на опросы консультантов из более чем 40 фирм, оказывающих услуги по менеджменту, ИТ, реинжинирингу, формулировке стратегий бизнеса, эксплуатации оборудования и т.п. К факторам, способствующим успеху реинжиниринга, отнесены:

Мотивация. Мотив для начала проекта по реинжинирингу должен быть ясно определен и зафиксирован. Высшее руководство должно быть абсолютно убеждено, что проект по реинжинирингу действительно даст значительный результат, и оно должно понимать, что этот результат существенно затронет некоторые структуры в компании. Чтобы обеспечить успех, **руководство должно верить в необходимость реинжиниринга**, проводимого в масштабах всей компании, и предоставить лучшие силы в распоряжение команды по реинжинирингу.

Руководство. Проект должен выполняться под управлением руководства компании. Руководитель, возглавляющий проект по реинжинирингу, должен иметь большой авторитет в компании и нести за него ответственность. Вероятность успеха выше у тех компаний, которые реалистично оценивают цели реинжиниринга и ожидаемые результаты, имеют представление о сроках, необходимых для его

реализации, усилиях и финансах. Для успеха проекта очень важно твердое и умелое управление.

До начала реинжиниринга руководитель должен отдавать себе отчет в трудностях, неизбежных при построении новой компании: он должен сопротивляться давлению упрямой старой компании и должен убедить людей в том, что проект не только выполним, но и необходим для выживания компании. Он не должен ни сдаваться, ни допускать упрощений, порочащих общую идею. Он должен наконец прилагать усилия для продвижения проекта вперед и для его своевременного завершения.

Сотрудники. В команде, выполняющей проект по реинжинирингу и контролирующей его выполнение, необходимо участие сотрудников, наделенных полномочиями и способных создать атмосферу сотрудничества. Люди должны понимать, почему проект приведен в действие (другими словами, они должны оценивать проблемы, которые мешают бизнесу), принимать свои новые задачи, быть способными выполнить их, посвящать реинжинирингу необходимое время и обоснованными шагами двигаться к успеху.

Относительно просто объяснить новый способ работы работникам нижнего уровня, но людям, исполняющим должности менеджеров, намного труднее понять, что новая компания может им предложить. Среди менеджеров наиболее важная группа - это средний уровень, который Б.Виллох [10] называет "постоянно замороженный".

Б.Виллох определяет три категории средних менеджеров: "тигры" - это молодые карьеристы, которые, хотя и участвуют с энтузиазмом в проекте по реинжинирингу, имеют тенденцию концентрироваться на своих собственных задачах в ущерб общим целям проекта; "ослы" - это старейшие сотрудники, достигнувшие целей своей карьеры, которые хотят спокойствия и стабильности в компании, они могут серьезно навредить проекту; "акулы" - сотрудники, которые разработали процедуры и инструкции для управления операциями компании, они часто имеют реальную силу в компании и могут создать огромные проблемы путем саботирования реальных перемен в жизни компании.

Проекты, сформулированные в терминах роста и расширения, а не сокращения продукции и расходов, имеют больше шансов на успех, поскольку порождают больше энтузиазма и меньше сопротивления. Люди с охотой объединяются вокруг важных инициатив, что редко случается, если речь идет только о вопросах стоимости.

Понятность (vision). Новые задачи компании должны быть четко сформулированы и понятны каждому сотруднику. Успешность реинжиниринга зависит от того, насколько руководство и рядовые сотрудники компании понимают, как достичь стратегических целей.

Бюджет. Проект должен иметь свой собственный бюджет, особенно если планируется интенсивное использование ИТ. Часто ошибочно считают, что реинжиниринг возможен на условиях самофинансирования.

Фокусирование. Работа по изменению компании должна фокусироваться на наиболее приоритетных целях, а ресурсы должны быть направлены именно на эти цели. (“Никогда не откусывайте больше, чем вы можете прожевать”).

Четко определенные роли и обязанности. Кроме людей, хорошо образованных в области реконструируемого бизнеса, необходимы люди, знающие, как изменять этот бизнес.

Осязаемые результаты. Результаты работ по реинжинирингу должны быть конкретными.

Технологическая поддержка. Для проведения работ по реинжинирингу необходима поддержка в форме методик и инструментальных средств. Инжиниринг бизнеса обычно включает в себя построение информационной системы для поддержки нового бизнеса (гл. 8). Это область риска, которая очень часто недооценивается. Специалисты по ИТ, в основном, не так компетентны в вопросе применения своих собственных продуктов (информационных систем), как специалисты в других инженерных дисциплинах (например, строители домов, мостов, автомобилей и электронных приборов). По данным университета Карнеги Меллон, 85 % всего программного обеспечения, разработанного в 1989 г., было произведено без использования каких-либо реальных методик и средств. Это приводит к большому количеству неудач.

Консультации. Эксперты (консультанты) могут оказать существенную помощь исполнителям, впервые осуществляющим реинжиниринг. Важно подчеркнуть, что консультанты выполняют поддерживающую, а не управляющую роль, и они не должны входить в штат компании.

Риск. Как сказал Дж.Карлсон [2], “нужно отважиться сделать прыжок”.

Естественно, что некоторые из этих предпосылок, например, стиль отношений в коллективе, создать быстро невозможно, поэтому работу над их реализацией необходимо начинать заранее.

1.3.2

Типичные ошибки при проведении реинжиниринга

Ранее отмечалось, что риск при проведении реинжиниринга довольно значителен. Однако необходимо подчеркнуть, что причины неудач заключены не в загадочности реинжиниринга, а в нарушении

правил его проведения. М. Хаммер и Дж. Чампи [5] указывают, что с точки зрения риска реинжиниринг подобен игре в шахматы, а не игре в рулетку, т.е. участники реинжиниринга, как игроки в шахматы, в меру своих знаний и умений могут влиять на результат. Главное - избегать глобальных ошибок.

Проект ожидает крах, если:

- ❶ ответственный за проект имеет недостаточно высокий должностной ранг;
- ❷ происходит излишняя концентрация на технологических вопросах;
- ❸ среди менеджеров преобладает отношение “сделайте это для меня”, т.е. они не вовлекаются в сам процесс.

Поскольку ИТ являются одним из ключевых элементов реинжиниринга, то разработчики проекта должны иметь представление о возможностях их использования в конкретном бизнесе. Например, технология сканирования документов, преобразования их в образы позволяет уйти от традиционных способов работы с документами в бумажно-интенсивных отраслях. К числу технологий, с которыми участники проектного исследования должны быть знакомы, относятся экспертные системы, системы управления большими базами данных, рабочие станции, интерактивные видеодиски и др. [1].

Идеи, принципы, обучение и консультации не гарантируют успех в инжиниринге бизнеса. Риск реинжиниринга бизнеса можно разделить на две категории:

- ❶ риск, связанный с изменением процесса;
- ❷ риск, связанный с используемой технологией.

По мнению большинства специалистов, 80% неудач при проведении реинжиниринга вызваны такими “мягкими” факторами, как мотивация, обязательность участия руководства, необходимость экспертного руководства. Для того чтобы гарантировать успех проекта по реинжинирингу, необходимо формализовать процесс реинжиниринга, что будет показано в последующих главах.

До недавнего времени проекты по реинжинирингу не имели поддержки в виде ориентированных на решение этих задач инструментальных средств; начиная с 1994 г., такие средства начали появляться (гл. 9). Однако в настоящее время мало интегрированных многофункциональных инструментальных средств, которые существенно снижают риск неудачи при проведении реинжиниринга.

Рассмотрим наиболее характерные ошибки, встречающиеся при проведении реинжиниринга.

❶ *Компания пытается улучшить существующий процесс вместо того, чтобы перепроектировать его.*

Это наиболее грубая ошибка, хотя довольно распространенная. Например, компания IBM Credit до получения описанного выше (см. разд. 1.2) положительного опыта проведения реинжиниринга, пред-

принимала несколько безуспешных попыток, которые по сути дела сводились к автоматизации существующих процессов, а не к их радикальному перепроектированию. Автоматизация не могла привести к успеху, так как она не меняла сути процесса, в ходе которого выполняемые экспертами вручную формы путешествовали из одного подразделения в другое. Более того, автоматизация не только не приносила значимого результата, она увековечивала плохо организованный процесс путем запоминания его в компьютере, что усложняло проведение успешных преобразований в будущем.

Не достигнув желаемых результатов на пути автоматизации, специалисты начинают применять разнообразные методики, направленные на улучшение деятельности компании. Пытаются применять теорию очередей и линейное программирование для того, чтобы сбалансировать работы между подразделениями и минимизировать время ожидания. Как правило, результаты не удовлетворяют. Начинают устанавливать стандарты по эффективности работы на каждом шаге процесса и т.д. и т.п.

Однако потерпев неудачи с автоматизацией и улучшениями, компании, как правило, все равно избегают радикальной перестройки своих процессов. Консерватизм объясняется тем, что существующие процессы понятны и поддерживаются соответствующей инфраструктурой и кажется, что инкрементное (частичное) улучшение старых процессов - наиболее безболезненный и безопасный путь.

Итак, для большинства компаний основной причиной неудач реинжиниринга является стремление к инкрементным улучшениям вопреки радикальной перестройке процессов.

② Компании не концентрируются на бизнес-процессах.

Не так давно американские филиалы ведущих фирм Европы создали исследовательские группы, цель которых состояла в том, чтобы за 90 дней разработать рекомендации по решению таких критических проблем, как инновации, обслуживание клиентов, полномочия (empowerment), рабочая команда (teamwork) и др. Несмотря на неограниченные полномочия, группы не смогли дать ничего кроме базовых рекомендаций.

Причина неудач состояла в том, что **исходная задача была плохо определена**. Используемые базовые понятия (инновации, полномочия, рабочая команда и др.) описывают характеристики и атрибуты сущностей, значимых для компаний, однако не существует непосредственного способа их достижения, так как они являются следствием выполняемых процессов и достигаются только в контексте этих процессов.

③ Компании концентрируются только на перепроектировании процессов, игнорируя все остальное.

Проведение реинжиниринга вызывает значительные изменения в таких областях, как проектирование работ, организационные структуры, системы управления и оценок и т.п. (см. разд. 2.4). Разнообра-

зие последствий приводит к тому, что даже менеджеры, заинтересованные в радикальном перепроектировании процессов, избегают проводить все требуемые изменения.

При проведении реинжиниринга часто повторяется следующий сценарий: менеджер верхнего уровня предлагает команде, осуществляющей реинжиниринг, совершить некоторое решительное преобразование существующего процесса. Некоторое время спустя команда, проанализировав последствия предложенного преобразования, показывает, что это ускорит время выполнения процесса в 10 раз, снизит его стоимость в 20 раз и в десятки раз уменьшит ошибки. Подобная информация радует менеджера, однако команда сообщает, что реализация нового процесса потребует новой системы оценки работ, объединения нескольких отделений, изменения стиля производственных отношений и т.п. Менеджер говорит: “Я просил вас сократить стоимость и ошибки, а не переделывать всю компанию”. Подобные рассуждения менеджера являются ошибочными, так как **реинжиниринг занимается именно “переделыванием” компании.**

④ Недооценка роли ценностей и убеждений исполнителей.

Для того чтобы исполнители эффективно выполняли перепроектированные процессы, они должны иметь побудительные причины. Недостаточно просто определить новый процесс, необходимо чтобы менеджеры сформировали и провели в жизнь новые системы ценностей и убеждений (см. разд. 2.4). Другими словами, менеджеры должны заботиться не только о том, что происходит на рабочих местах исполнителей, но и о том, что происходит в их головах.

Когда в компании Ford провели реинжиниринг способов расчета с поставщиками, потребовалось изменить отношение сотрудников Ford к поставщикам: поставщики теперь должны были восприниматься не как конкуренты, а как партнеры Ford по данному бизнес-процессу.

Изменения в позиции исполнителей достигаются не так легко. Недостаточно просто произнести речь перед исполнителями. Требуется новая система управления, которая культивировала бы требуемые ценности и вознаграждала за их поддержание. **Менеджеры должны не только произносить соответствующие речи о новых ценностях, но и подтверждать их соблюдение своим поведением.**

⑤ Согласие довольствоваться малым.

Значительные результаты достигаются только при больших амбициях руководства компании. При проведении реинжиниринга встречается такая ситуация, когда кто-то из менеджеров предлагает: “лучше синица в руках, чем журавль в небе”, т.е. обещает получить без больших затрат и без перестройки, свойственной реинжинирингу, повышение эффективности работ на 10-20 %. Искушение выбрать более легкий путь, сводящийся к усовершенствованиям, оказывается

довольно соблазнительным. Однако усовершенствования, как правило, усложняют существующий процесс, а их наслоения делают процесс малопонятным. Более того, усовершенствования требуют затрат времени и денег на существующий неэффективный процесс, усложняют управление и приводят к “развитию культуры инкрементализма, лишаящей компанию мужества и доблести” [5].

6 *Преждевременное завершение реинжиниринга.*

Не вызывает удивления, что существуют компании, которые отказываются от проведения реинжиниринга при появлении первых трудностей. Удивительно, что существуют компании, которые сворачивают деятельность по реинжинирингу при достижении первых успехов. Начальный успех становится предлогом для возврата к более привычному способу ведения бизнеса.

7 *Ограниченная постановка задачи.*

Реинжиниринг обречен на неудачу до начала работ, если ограничена область его действия или задача поставлена слишком узко. Реинжиниринг начинается с определения целей, которые должны быть достигнуты, а не с определения способов их достижения.

Поясним на следующем примере. Старший менеджер компании Х, производящей оборудование, считает, что процесс оформления заказов клиентов является чрезмерно дорогим и просит команду, проводящую реинжиниринг, снизить стоимость этого процесса. Команда исследует проблему и приходит к выводу, что стоимость оформления заказов можно существенно снизить, если компания будет продавать клиентам только ту продукцию, которую она сама производит. Формально вывод команды решает поставленную задачу, хотя скорее всего не устроит клиентов. Дело в том, что задача была поставлена слишком узко, так как нельзя, не нарушая всего процесса обслуживания клиента, говорить об изменении процесса оформления заказов.

Довольно часто встречается такая ситуация. Компания определяет некоторый процесс, который она хочет перестроить. Однако как только реинжиниринг начинается, вместо всего процесса рассматривается только его некоторый фрагмент, так как существующие организационные границы не позволяют охватить весь процесс. Необходимо помнить, что задача реинжиниринга не укреплять, а разрушать существующие организационные границы.

8 *Существующая корпоративная культура и принятые в компании принципы управления могут препятствовать реинжинирингу.*

Культура компании может не позволить даже начать реинжиниринг. Например, если компания принимает решения на основании консенсуса, то сотрудники этой компании могут посчитать принцип проведения реинжиниринга “сверху-вниз” оскорбляющим их чувства. Компании с короткой историей существования обычно свято

чтут необходимость гарантировать ежеквартальные результаты, поэтому эти компании могут считать преобразования, связанные с реинжинирингом, весьма рискованными.

⑨ *Попытки осуществлять реинжиниринг не “сверху-вниз”, а “снизу-вверх”.*

Реинжиниринг никогда не проводится “снизу-вверх”, он всегда проводится “сверху-вниз”. Существует две причины, по которым реинжиниринг не может быть успешно завершен менеджерами нижнего и среднего уровня.

Первая причина состоит в том, что менеджеры нижнего и среднего уровня не обладают той широтой взглядов на деятельность компании, которая необходима для проведения реинжиниринга. Их опыт в основном ограничивается знанием тех функций, которые они выполняют в своем подразделении. Они, как правило, лучше других осознают узкие проблемы своего подразделения, но им трудно увидеть процесс в целом и распознать его слабые места. Менеджеры среднего и нижнего уровня успешно осуществляют инкрементные улучшения, а не реинжиниринг.

Вторая причина состоит в том, что бизнес-процессы неизбежно пересекают организационные границы, т.е. границы подразделений, поэтому менеджеры нижнего и среднего уровня не имеют достаточного авторитета для того, чтобы настаивать на трансформации процессов. Более того, радикальные преобразования существующего процесса могут привести к уменьшению влияния и авторитета того или иного менеджера среднего уровня. Менеджеры среднего уровня обычно много вложили в существующий способ выполнения процесса, и их будущее, вызванное перестройкой процесса, может оказаться неочевидным. По этим причинам менеджеры среднего уровня могут не только не способствовать проведению реинжиниринга, но препятствовать ему.

⑩ *Назначение ответственным за реинжиниринг старшего менеджера, который не понимает, что это такое.*

Для успеха реинжиниринга недостаточно назначить руководителем старшего менеджера. Необходимо, чтобы он понимал, что такое реинжиниринг, был предан ему и мог думать в терминах процессов.

⑪ *Недостаточное выделение ресурсов на проведение реинжиниринга.*

Достижение существенного повышения эффективности деятельности компании, являющегося следствием реинжиниринга, невозможно без существенных инвестиций в программу проведения реинжиниринга. Наиболее важный компонент этих инвестиций заключается в затратах времени и работы наиболее ответственных людей компании.

Реинжиниринг требует прямой и персональной ответственности руководства компании. Ответственность не может быть делегирована вниз. Старшее руководство компании не должно самостоятельно осуществлять реинжиниринг, оно может иметь помощников и соратников, но оно не должно перекладывать ответственность за проведение реинжиниринга на своих подчиненных.

Выделение недостаточных ресурсов на проведение реинжиниринга сигнализирует руководству компании, что кто-то не осознает важности проведения реинжиниринга и сопротивляется его проведению.

12 *Реинжиниринг проводится на фоне множества других мероприятий.*

Если руководство компании не уделяет реинжинирингу основное внимание, то проведение реинжиниринга обречено на неудачу. Реинжиниринг не должен осуществляться на фоне других программ и мероприятий.

13 *Количество проектов по реинжинирингу не должно быть большим.*

Компания не должна одновременно осуществлять реинжиниринг большого количества процессов, так как время и внимание управленческого аппарата ограничено, а при проведении реинжиниринга недопустимо, чтобы внимание менеджеров непрерывно переключалось между различными проектами.

14 *Нецелесообразно проводить реинжиниринг за год или за два до отставки исполнительного директора компании.*

Шансы на успешное проведение реинжиниринга заметно снижаются, если известно, что исполнительный директор компании через год или два уходит в отставку. Дело здесь не в том, что исполнительный директор не будет заботиться о будущем компании или будет недостаточно старателен. Дело в том, что проведение реинжиниринга неизбежно повлечет за собой изменения в структуре компании и в ее управляющих системах. В этих условиях исполнительный директор может не захотеть брать обязательства, которые будут стеснять его преемника.

Кроме того, последствия реинжиниринга будут влиять на других менеджеров, особенно в иерархически организованных компаниях. Претенденты на ведущий пост в компании неизбежно осознают, что за ними наблюдают и их оценивают. В этих условиях возникает соблазн больше заботиться о личных показателях, а не о коллективной работе по успешному завершению реинжиниринга. Более того, претенденты не заинтересованы в проведении любых преобразований, которые ущемляют их позиции в компании.

15 *Компания концентрируется исключительно на замыслах.*

Реинжиниринг не сводится к замыслам, необходимо обеспечить их реализацию. Различие между победителями и проигравшими при проведении реинжиниринга состоит не в качестве использованных идей, а в том, были ли эти идеи реализованы.

16 *Попытка провести реинжиниринг, не ущемив ничьих интересов.*

Афоризм о том, что нельзя приготовить омлет не разбив яйца, весьма точно отражает суть реинжиниринга. Реинжиниринг не приносит всем только радости. В результате реинжиниринга одним сотрудникам приходится изменять характер своей работы, другие сотрудники могут потерять работу, третьи - будут чувствовать себя не очень комфортно после проведения реинжиниринга. Стремление угодить всем невыполнимо, что вызывает либо откладывание реинжиниринга, либо ведет к замене его на последовательность инкрементных изменений.

17 *Компания отступает, когда встречает сопротивление сотрудников, недовольных последствиями реинжиниринга.*

То, что некоторые сотрудники компании будут сопротивляться изменениям, вызванным реинжинирингом, не должно удивлять никого и особенно руководителей проекта по реинжинирингу. Сопротивление - естественная реакция на существенные перемены. Иногда сопротивление ошибочно рассматривается как первопричина неудач реинжиниринга. Это заблуждение. Нельзя же утверждать, что причиной автомобильных аварий является закон Ньютона. Причиной аварий являются водители, которые не учитывают этот закон при вождении автомобилей. Аналогично не реинжиниринг, а ошибки в управлении, которые не позволили предвидеть и учесть неизбежное сопротивление преобразованиям, являются первопричиной неудач.

18 *Растянутое проведение реинжиниринга.*

Проведение реинжиниринга создает определенное напряжение в компании, и затягивать этот процесс весьма опасно. Опыт проведения реинжиниринга показывает, что 12 месяцев обычно достаточно для того, чтобы компания прошла путь от декларации идей до завершения первой действующей версии реконструированных процессов. Затраты большего времени приводят к тому, что сотрудники компании становятся нетерпеливыми, встревоженными и сбитыми с толку. Они начинают думать, что реинжиниринг есть очередная фальшивая программа преобразований.

Завершая перечисление наиболее распространенных ошибок, допускаемых при проведении реинжиниринга, отметим, что существует множество примеров успешного проведения реинжиниринга. Компания, руководители которой понимают основы реинжиниринга и привержены этой идее, имеет практически стопроцентный шанс добиться успеха. Выигрыш от успешного проведения реинжиниринга - это не только ярчайшее событие в жизни фирмы, но значимое собы-

тие для страны, учитывая огромное влияние, которое окажет на экономику эффективная перестройка важнейших компаний, организаций, акционерных обществ и банков.

Время для колебаний и сомнений закончилось, началось время действий.

1.4

Что не является реинжинирингом

Люди, знающие о реинжиниринге понаслышке или только осваивающие его основные понятия, часто путают их с другими известными им понятиями. Например, они могут сказать: “Реинжиниринг - это то же самое, что и downsizing (уменьшение размерности)”. Как правило, реинжиниринг мало или ничего общего не имеет с этими понятиями. Так что же не является реинжинирингом?

❶ Несмотря на важную роль, которую информационные технологии играют в БПР, *реинжиниринг не то же самое, что автоматизация или автоматизация бизнес-процессов* - business process automation (BPA). BPA с помощью информационных технологий автоматизирует существующий процесс (со всеми его недостатками) и не ставит перед собой основную задачу реинжиниринга - проектирование **н о в о г о** процесса, благодаря чему достигаются основные результаты. В связи с этим автоматизация не приводит к существенному улучшению показателей деятельности компании (улучшения в диапазоне 10-50 %). **Задача реинжиниринга - фундаментальное переосмысление и радикальное перепроектирование существующей работы (процессов) для получения резких (скачкообразных) изменений (улучшения в диапазоне 500 - 1000 и более процентов).**

❷ Некоторые пользователи путают реинжиниринг бизнеса с реинжинирингом программного обеспечения (software reengineering), задача которого - переписать с помощью современных технологий устаревшие информационные системы, как правило, не изменяя сути автоматизируемых процессов.

❸ Реинжиниринг *не является ни реструктурированием (restructuring), ни уменьшением размерности (downsizing)*. Эти термины обозначают уменьшение возможностей для соответствия текущим более низким требованиям. Например, на рынке падает спрос на автомобили компании GM, и эта компания должна быть способна перестроить производство (с минимальными затратами) на уменьшение количества выпускаемых автомобилей. **Реинжиниринг в отличие от реструктурирования и уменьшения размерности стремится сделать больше, а не меньше.**

④ Реинжиниринг не является ни реорганизацией (*reorganizing*), ни “выравниванием организационной структуры” (*flattening*), хотя “выравнивание” может быть результатом проведения реинжиниринга. В отличие от реорганизации и выравнивания, имеющих дело с организационными структурами, реинжиниринг имеет дело с процессами.

Многие компании видят причину своих бед в бюрократии и пытаются с ней бороться. Это ошибочная точка зрения. При традиционной иерархической структуре компании, где процессы разбиты на фрагменты, выполняемые в подразделениях, без бюрократии наступит хаос. Бюрократия - это клей, соединяющий вместе подразделения традиционной организации. Избавиться от бюрократии можно только с помощью реинжиниринга, который позволит не разбивать процессы на фрагменты.

⑤ Реинжиниринг не является ни улучшением качества (*quality improvement - QI*), ни тотальным управлением качеством (*total quality management - TQM*). Реинжиниринг и управление качеством имеют совпадающие точки зрения на важность процессов. Однако управление качеством за основу принимает существующие процессы, которые требуется улучшить (цель: делать то, что делали, но только лучше), в то время как реинжиниринг заменяет существующие процессы на новые.

1.5

Ответы на типичные вопросы о реинжиниринге

В период с 1993 г. до конца 1996 г. в мире опубликованы десятки монографий и сотни статей по реинжинирингу бизнес-процессов, в США прошли две национальные конференции по БПР. Этот обширный материал позволяет сформулировать типичные вопросы, которые читатели и слушатели задают специалистам по БПР. Приведем наиболее интересные вопросы и возможные ответы на них.

① Почему велика вероятность неудачи при выполнении реинжиниринга?

В первой монографии, посвященной БПР [5], было сказано, что от 50% до 70% проектов по реинжинирингу заканчиваются неудачами. Авторы во втором издании этой монографии подчеркивают, что указанный процент неудач является следствием о ш и б о к, совершаемых при проведении реинжиниринга (перечень ошибок, приводящих к неудачам, приведен в разд. 1.3), а не следствием некоей мис-

тической непредсказуемости БПР. Авторы указывают, что БПР подобен игре в шахматы, где выигрывает тот, кто лучше умеет играть.

② Не является ли реинжиниринг очередным проходящим увлечением в менеджменте?

Можно привести два довода за то, что реинжиниринг не является временным увлечением. Во-первых, БПР - не чудодейственное лекарство, проведение реинжиниринга требует напряженной работы. Реинжиниринг не сводится к одной узконаправленной методике для решения всех проблем, реинжиниринг - это интегрированная совокупность различных подходов, влекущих за собой радикальное переосмысление всех аспектов бизнеса. Во-вторых, ведущие компании всего мира при правильном применении реинжиниринга достигают беспрецедентного увеличения эффективности своей деятельности.

③ В чем различие между реинжинирингом и TQM (total quality management)?

Реинжиниринг и TQM не идентичны, но и не противоречат друг другу, они дополняют друг друга: оба фокусируются на клиентах и на процессах, но между ними есть важные различия. Реинжиниринг предназначен для проведения радикального перепроектирования компании, обеспечивающего существенное повышение эффективности, в то время как TQM ориентирован на достижения постепенного улучшения. Оба подхода имеют право на существование. Реинжиниринг используется периодически для радикальных преобразований, а в период между ними применяется TQM для настройки процессов.

Кроме того, TQM встраивается в культуру компании и может осуществляться без ежедневного внимания со стороны руководства верхнего звена. Реинжиниринг, наоборот, требует интенсивного, инициируемого сверху воздействия руководства компании.

④ Изменилось ли понимание термина “реинжиниринг” со времени его первого появления?

Определение реинжиниринга, данное М.Хаммером [5], содержит четыре ключевых слова: *фундаментальный, радикальный, резкий (скачкообразный) и процесс*. Первоначально М.Хаммер считал, что наиболее важным из этих слов является “радикальный”. В настоящее время большинство специалистов (в том числе и М.Хаммер) пришли к выводу о том, что наиболее важно слово “процесс”, так как при реинжиниринге управление бизнесом всегда сводится к процессам.

Кроме того, задачи реинжиниринга по-разному толковались в различные периоды эволюции БПР (см. разд. 2.1).

⑤ “Реинжиниринг” изобретен учеными?

Нет. Реинжиниринг не изобретен, а обнаружен учеными в ряде преуспевающих компаний, которые интуитивно использовали его основные свойства. Ученые выделили ос-

новые характерные черты, систематизировали и сформулировали определение реинжиниринга.

6 В каких областях получен наибольший опыт проведения реинжиниринга?

Раньше других (в 1992-1993 гг.) реинжиниринг начал применяться в страховании, в телекоммуникации и энергетике, затем он проник в химию, электронику, вычислительную технику, производство товаров широкого потребления. С 1994 г. началось активное внедрение реинжиниринга в банки и правительственные учреждения. Так, правительство США с 1994 г. организовало более 200 проектов по реинжинирингу, а 75% из 400 крупнейших североамериканских банков увеличивают число проектов по реинжинирингу, хотя и так среднее число проектов по БПР в 1994 г. составило 3,6 на банк. 100 крупнейших банков США планируют затратить на реинжиниринг в 1997 г. 3,9 млрд. дол. [8].

7 Применяется ли реинжиниринг в общественном секторе (в правительственных и общественных организациях)?

Да, безусловно [6]. Президент США Б.Клинтон в своей первой предвыборной кампании много внимания уделял реинжинирингу правительственных организаций. После своего избрания на пост президента Б.Клинтон поручил вице-президенту вести работы по реинжинирингу правительственных организаций. В 1994 г. в США начато более 200 проектов по реинжинирингу. Однако необходимо подчеркнуть, что при реинжиниринге правительственных и общественных организаций возникают некоторые специальные проблемы:

- Для компаний общественного сектора трудно предложить способ оценки эффективности их функционирования. Критерием оценки деятельности компаний частного сектора является прибыль, которая может быть естественным образом связана с другими показателями деятельности компании. Деятельность большинства организаций общественного сектора может быть оценена только по показателю стоимости (а не прибыли), что затрудняет сопоставление показателей деятельности компании.

- Разрушение барьеров между подразделениями частной компании намного легче, чем разрушение барьеров между правительственными организациями. Проблема в том, что функции, например, федерального правительства затрагивают не одну организацию, а сеть организаций. Например, при въезде в США иностранец должен иметь дело с тремя независимыми организациями: иммиграции, натурализации и сельскохозяйственной инспекции. Поэтому проведение реинжиниринга в одной из трех организаций не оказывает существенного влияния на общее время взаимодействия путешественника с правительственными службами.

8 *Реинжиниринг целесообразно применять только к крупным компаниям?*

При возникновении реинжиниринга многие специалисты считали, что его целесообразно применять к крупным компаниям с многомиллиардным оборотом. Однако опыт показал, что реинжиниринг с успехом применяется и к отдельным подразделениям крупных компаний, и к совсем маленьким компаниям.

9 *Реинжиниринг применяется только в Америке?*

Нет. Хотя компании Америки были в авангарде этого движения, реинжиниринг быстро распространился по всему миру. Наибольший интерес БПР вызвал, кроме Америки, в Бразилии, Корее, в странах Латинской Америки и Восточной Азии.

10 *Какое количество процессов целесообразно перепроектировать на начальной фазе реинжиниринга?*

Основная проблема не в количестве процессов, а в том, какие процессы перепроектировать. Перепроектирование большого количества подчиненных процессов не принесет ощутимого результата, необходимо проводить реинжиниринг основных процессов компании. Проведение реинжиниринга как малых, так и больших (основных) процессов является довольно трудной задачей, так как при этом неизбежно затрагиваются функции подразделений и исполнителей, что может приводить к хаосу.

11 *Может ли реинжиниринг иметь успех в компаниях с мощными профсоюзами?*

Реинжиниринг не затрагивает интересы рабочих - наиболее активной профсоюзной прослойки. Последствия реинжиниринга уменьшают роль менеджеров среднего звена. Тем не менее проведение реинжиниринга может вызвать напряженность среди членов профсоюза, особенно если ему предшествовало сокращение. Хотя в результате проведения реинжиниринга некоторые сотрудники могут потерять работу, цель реинжиниринга - реорганизовать производственные процессы, а не сократить сотрудников. Однако для того чтобы предотвратить возможную оппозицию реинжинирингу, необходимо максимально рано вовлекать сотрудников в этот процесс. Профсоюзное руководство компании обычно привлекается к участию в процессе реинжиниринга с самого его начала.

ЛИТЕРАТУРА

1. Barrett J. L. Process visualisation. Getting the vision right is key // Information systems management. - 1994. - V. 11 - № 2.
2. Carlzon J. Moments of Truth. MA: Ballinger Publishing Company. 1987.
3. Davenport T. H. Business Innovation, Reengineering Work through Information Technology. - Boston: Harvard Business School Press, 1993.

4. **Flynn K. Critical Success Factors for a Successful Business Reengineering Project** //CASE World Conference Proceedings. - Boston: 1993, October.
5. **Hammer M. and Champy J. Reengineering the Corporation: A Manifesto for Business Revolution.** - N-Y: HarperCollins, 1993.
6. **Harmon P. Business Process Reengineering with Objects - Part 2** //Object-Oriented Strategies. - 1995. - V. 5. - № 1.
7. **Jacobson I., Ericsson M., Jacobson A. The Object Advantage: Business Process Reengineering with Object Technology.** ACM Press. - Addison-Wesley Publishing, N-Y: 1995.
8. **Kleinberg. BPR Tools Categories - Multiple Choices**//Research Note: July 7. 1995. GartnerGroup.
9. **Martin J. Enterprise Engineering - The Key to Corporate Survival.** - V. I-V. - UK: Savant Institute, 1994.
10. **Willoch B. E. Business Process Engineering. Vinnende arbeidsprosesser og organisasjonsstrukturere i forandringens decennium.** - Norway: Fagbogforlaget, 1994.

2

РЕИНЖИНИРИНГ И ЕГО ВОЗДЕЙСТВИЕ НА КОМПАНИЮ

2.1

Инжиниринг бизнеса, реинжиниринг бизнеса и усовершенствование бизнеса

Специалисты рассматривают инжиниринг бизнеса как общее понятие, включающее реинжиниринг бизнес-процессов и усовершенствование бизнеса:

ИНЖИНИРИНГ БИЗНЕСА = РЕИНЖИНИРИНГ БИЗНЕСА + УСОВЕРШЕНСТВОВАНИЕ БИЗНЕСА

Хотя эта книга посвящена реинжинирингу бизнеса, ниже будут кратко рассмотрены особенности реинжиниринга и усовершенствования бизнеса. В настоящее время основное внимание уделяется реинжинирингу бизнеса, потому что именно он дает наиболее впечатляющие и кардинальные результаты (разумеется, в случае успеха). Усовершенствование бизнеса, которое имеет дело с долговременной поддержкой перепроектированных процессов, не так бросается в глаза в основном потому, что компании и так постоянно им занимаются.

Реинжиниринг бизнеса подразумевает, что осуществлен исчерпывающий анализ существующего бизнеса, проанализировано, почему вы делаете то, что вы делаете и как вы это делаете и т.д. Задача БПР - попытаться найти совершенно новый способ реконструирования существующего бизнеса (или, по крайней мере, его наиболее важных процессов), используя новые технические достижения (например, современные информационные технологии) для лучшего обслуживания своих клиентов.

Ряд авторов, включая Х.Юхансона [6] и Б.Виллоха [8], называют это реинжинирингом бизнес-процессов (а не реинжинирингом бизнеса). Т.Давенпорт [3] использует термин "инновация процессов", оп-

ределяя его как “существенное снижение стоимости процесса или сокращение времени, или значительное улучшение качества, гибкости, уровня обслуживания или других параметров бизнеса”.

Ряд специалистов считает, что слово “процесс” в термине “реинжиниринг бизнес-процессов” можно удалить как избыточное, так как в будущем реинжиниринг бизнеса будет означать только реинжиниринг бизнес-процессов и ничего более. Как правило, реинжиниринг может осуществляться много раз до тех пор, пока большинство процессов компании не будет полностью перепроектировано. Работа таким образом подразделяется на фазы, каждая из которых имеет ясно определенную цель. Наиболее важная цель - достигнуть существенных улучшений в работе компании, измеряемых в критических параметрах процессов.

После того как компания провела реинжиниринг своих процессов, они должны поддерживаться и улучшаться, что требует новых целей и новых усилий для их достижения. Обычно эти цели значительно скромнее, чем цели реинжиниринга, и работа, требующаяся для их достижения, не имеет значительного влияния на эффективность работы компании. Эта работа по изменению компании локальна и не охватывает весь бизнес. Однако усовершенствование бизнес-процессов должно продолжаться, так как это очень важно для всей компании в “беге на длинную дистанцию” [5].

Т.Давенпорт [3] для того, чтобы описать различие между усовершенствованием и реинжинирингом бизнеса, использовал следующую таблицу (табл. 2.1).

Таблица 2.1

Различия между усовершенствованием и реинжинирингом бизнеса

Наименование параметра	Усовершенствование	Реинжиниринг
Уровень изменений	Наращиваемый	Радикальный
Начальная точка	Существующий процесс	“Чистая доска”
Частота изменений	Непрерывно/единовременно	Единовременно
Требуемое время	Короткое	Длительное
Направление	Снизу-вверх	Сверху-вниз
Охват	Узкий, на уровне функций	Широкий, межфункциональный
Риск	Умеренный	Высокий
Основное средство	Статистическое управление	Информационные технологии
Тип изменения	Культурный	Культурный/структурный

Как показывает таблица, реинжиниринг бизнес-процессов осуществляется один раз в течение нескольких лет. После завершения реинжиниринга компания становится объектом для усовершенствования. Таблица также показывает, что усовершенствование осуществляется внутри одного подразделения (функции), в то время как реинжиниринг касается всей компании. Большинство каждодневных усовершенствований выполняется отдельными исполнителями в конкретных подразделениях, т.е. усовершенствование выполняется снизу-вверх, в то время как реинжиниринг может быть реализован только сверху-вниз, т.е. по указанию и под управлением высшего руководства фирмы.

Другой интересный аспект таблицы заключается в том, что при усовершенствовании используются средства статистического управления, объясняющие и минимизирующие источники изменений. Эти средства не могут быть приспособлены к большим изменениям, характерным для реинжиниринга. Поэтому при реинжиниринге используются более мощные средства - информационные технологии.

Проект по реинжинирингу бизнеса обычно включает следующие четыре этапа (см. более детально в гл. 4):

Разработка образа (vision) будущей компании. На этом этапе компания строит картину того, как следует развивать бизнес, чтобы достичь стратегических целей.

Анализ существующего бизнеса. Проводится исследование компании и составляются схемы ее функционирования в настоящий момент.

Разработка нового бизнеса. Разрабатываются новые и (или) измененные процессы и поддерживающая их информационная система. Выполняется прототипирование и тестирование новых процессов.

Внедрение (installing) нового бизнеса. На этом этапе новый проект внедряется в бизнес.

Необходимо подчеркнуть, что перечисленные этапы выполняются не последовательно, а по крайней мере частично параллельно, причем некоторые этапы повторяются.

Реинжиниринг как любое и новое понятие претерпевает определенные изменения во времени. В настоящее время выделяют следующие периоды эволюции БПР [7].

❶ **Начальный (тактический) период: 1990 - 1993 гг.** На этом периоде в понятие БПР включали перепроектирование одного или небольшого количества значимых процессов. Реинжиниринг сопровождался такими программами усовершенствования бизнеса как:

- управление качеством;
- функционально-стоимостной анализ;
- производство "как раз вовремя" (just-in-time manufacturing).

Результатом БПР на этом периоде является значительное повышение эффективности работы компании.

② **Стратегический период: 1994 - 1996 гг.** На этом периоде в БПР включают реинжиниринг всех или большинства бизнес-процессов, однако в рамках существующей бизнес-стратегии. Результатом является скачкообразное повышение эффективности работы компании.

③ **Глобальный период: 1996 - 2000 гг.** Предполагается, что на этом периоде эволюции механизмы реинжиниринга обеспечат возможность постоянного пересмотра бизнес-стратегии компании в соответствии с изменяющимся внешним окружением.

2.2

Роль информационных технологий в реинжиниринге

В определении реинжиниринга делового процесса ключевым является слово “процесс” в контексте “фундаментальное переосмысление и радикальное перепроектирование деловых процессов” (см. разд. 1.1). Перепроектирование процессов становится возможным, как правило, благодаря использованию ИТ.

М.Хаммер и Дж.Чампи [4] выразили значимость ИТ для реинжиниринга в виде следующих утверждений.

① Компания, которая не может изменить свое мышление с дедуктивного на индуктивное (см. ниже), не готова к проведению реинжиниринга.

② Компания, которая ставит знак равенства между технологией и автоматизацией, не готова к проведению реинжиниринга.

③ Компания, которая сначала ищет проблемы, а затем для их решения подыскивает технологии, не готова к проведению реинжиниринга.

Эти авторы считают: для того, чтобы полностью использовать возможности информационных технологий, необходимо обучить менеджеров компании мыслить индуктивно, а не дедуктивно, как они привыкли (по мнению авторов данной книги, М.Хаммер и Дж.Чампи используют термины “дедуктивный и индуктивный” не в строгом смысле, однако суть их рассуждений верна). Итак, по М.Хаммеру и Дж.Чампи *дедуктивное мышление*, традиционно используемое менеджерами, состоит в следующем: менеджеры определяют проблему (проблемы), а затем ищут и оценивают различные способы решения этой проблемы. Однако при реинжиниринге надо использовать *индуктивное мышление*, т.е. способность сначала распознать эффективное решение, а затем искать проблемы, которые оно может разрешить.

Действительно во всех примерах успешного проведения реинжиниринга, рассмотренных нами в разд. 1.2, компании первоначально решали одну проблему, а затем понимали, что для успеха необходимо переформулировать исходную проблему. Компания Ford первоначально видела проблему так: найти метод более быстрой обработки накладных, требующий при этом меньше людей, но решение состояло в том, чтобы вообще отказаться от накладных. Компания IBM Credit первоначально видела проблему так: ускорить передачу информации между специалистами различных групп, но решение состояло в том, чтобы заменить всех специалистов одним человеком, имеющим доступ к экспертной системе с базой данных, т.е. отказаться от передачи информации между специалистами.

Основная ошибка большинства компаний в том, что они рассматривают ИТ через призму существующих процессов. Они формулируют проблему следующим образом: “Как использовать новые технологии, чтобы улучшить то, что мы сейчас делаем?”, т.е. компании хотят решить свои проблемы, автоматизируя существующую деятельность. Однако оказывается, что простое наклеивание ИТ на существующие деловые процессы не приводит к истинной трансформации бизнеса и даже наоборот может блокировать процесс перестройки, сохранив прежние способы ведения дел. Компании должны формулировать проблему иначе: “Как с помощью новой технологии делать то, что мы еще не делали?”

Реинжиниринг - это использование самых последних информационных технологий для достижения совершенно новых деловых целей.

Например, ценность мини-компьютеров, а затем и персональных компьютеров заключается не в том, что они делают то же, что и большие машины, а в том, что они породили целый класс новых приложений. Аналогичная ситуация с телеконференциями. Ожидалось, что они приведут к снижению числа командировок и экономии дорожных расходов. Однако на самом деле их воздействие в этой области оказалось незначительным, поскольку необходимость в личном контакте преобладает. Ценность телеконференций в том, что они меняют стиль работы, а не уменьшают ее стоимость и дают эффект при использовании в целях обучения, для проведения коротких еженедельных совещаний. Естественно, что ИТ играют свою роль в изменении деловых процессов наряду с другими важнейшими факторами, определяющими стратегию развития бизнеса, - администрированием, ценовой политикой, технологическими инновациями и т.п.

Решающая роль ИТ в реинжиниринге, обеспечивающая получение весомых конкурентных преимуществ, была проиллюстрирована в разд. 1.2 на примере опыта трех компаний: IBM Credit, Ford и Kodak. IBM Credit, благодаря реинжинирингу, достигла более чем 90% улучшения, в то время как автоматизация существующих про-

цессов позволяет получить 10%-ное улучшение. Ford, проведя реинжиниринг, уменьшил количество сотрудников на 80%, в то время как автоматизация позволяла получить только 20%-ное уменьшение. Фирма Kodak, благодаря реинжинирингу, почти вдвое уменьшила общий срок разработки нового продукта.

Приведем еще один пример. Компания Дженерал Моторс (ДМ) построила завод Сатурн (Спринг Хилл, Теннесси, США), в котором производственная база данных доступна в режиме on-line для официальных поставщиков. Теперь поставщики не ждут от Дженерал Моторс официального заказа на комплектующие. Они просматривают график работ и сами определяют, когда и сколько комплектующих должно быть поставлено. Зная, сколько машин компания выпускает в месяц, поставщик может составить собственный график производства и доставки соответствующих деталей. В этом процессе нет места бумажной переписке - заказам и счетам. Когда необходимые детали прибывают в нужное место и в строго определенное время, клерк сканирует нанесенный на них штриховой код, что автоматически подтверждает приемку товара и инициирует оплату. Таким образом, ИТ, а конкретнее, база данных по графику производства и электронный обмен данными позволили разным компаниям действовать как единое целое.

М.Хаммер и Дж.Чампи приводят примеры (табл. 2.2) того, как новые ИТ меняют старые правила работы компании [4].

Таким образом, современные ИТ позволяют радикально изменить бизнес-процессы и значительно улучшить основные показатели деятельности компании, что позволяет опередить конкурентов. Современные технологии продолжают развиваться, и поэтому те правила бизнеса, которые кажутся незыблемыми сегодня, могут устареть через год или ранее.

Необходимо понимать, что если Вы можете купить некоторую технологию, то это не самая современная технология. Действительно, до того, как технология становится доступной для покупателя, она апробируется разработчиком, т.е. он использует эту технологию для решения реальных задач. Компания не может сегодня “увидеть” технологию, а завтра начать ее использовать. Требуется время на изучение, на понимание того, где эту технологию целесообразно использовать, на популяризацию технологии внутри компании и на разработку плана ее внедрения. В общей сложности на это уходит около трех лет.

На основании вышеизложенного становится понятно, что роль ИТ в проведении реинжиниринга бизнес-процессов трудно переоценить, однако ошибочно думать, что ИТ - единственный существенный элемент для реинжиниринга.

Новые ИТ, изменяющие правила работы компаний

Прежнее правило	Технология	Новое правило
Информация может появляться в одно время в одном месте	Распределенные базы данных	Информация может появляться одновременно в разных местах тогда, когда она необходима
Сложную работу могут выполнять только эксперты	Экспертные системы	Работу эксперта может выполнять специалист по общим вопросам
Необходимо выбирать между централизацией и децентрализацией	Телекоммуникационные сети	Можно одновременно получать преимущества от централизации и децентрализации
Все решения принимают менеджеры	Средства поддержки принятия решений, доступ к БД, средства моделирования	Принятие решений становится частью работы каждого сотрудника (иерархическое принятие решения)
Специалистам для получения, хранения, поиска и передачи информации требуется офис	Беспроводная связь и переносимые компьютеры	Специалисты могут посылать и получать информацию из того места, где они находятся
Лучший контакт с потенциальным покупателем - личный контакт	Интерактивный видеодиск	Лучший контакт с потенциальным покупателем - эффективный контакт
Для того чтобы найти некую сущность, необходимо знать, где она находится	Технология автоматического индексирования и отслеживания	Сущности говорят вам, где они находятся
План пересматривается периодически	Высокопроизводительные ЭВМ	План пересматривается оперативно, по мере необходимости

В процессе реинжиниринга отделы ИТ в фирмах и корпорациях вынуждены пересматривать свою роль. Этому способствуют три фактора, которые в полной мере проявились в конце 90-х гг. Во-первых, бизнес стал оказывать все большее давление на отделы ИТ в плане увеличения их вклада в общий результат деятельности компании с целью создания и поддержания конкурентоспособности. Во-вторых, компьютерная парадигма вычислений, ориентированная на хосты, которая доминировала последние 30 лет, приходит в упадок и заменяется новой парадигмой - сетевыми вычислениями, что приводит к появлению ряда новых ИТ. В-третьих, растет конкуренция внутренним отделам ИТ со стороны внешних фирм, специализирующихся на информационных услугах. Компании, оказывающие такие услуги, утверждают, что они лучше подготовлены для удовлетворения потребностей клиентов в информационно-технологическом обслуживании. До тех пор пока внутренние ИТ подразделения не убедят управляющих, что вносят свою лепту в бизнес, фирмы, специализирующиеся на управлении информационными ресурсами клиента, могут выигрывать у внутренних отделов эту работу в открытой и свободной конкуренции.

Миссия отделов ИТ смещается от обслуживающих функций к формированию основ конкурентоспособности компании. Для реализации этой новой роли руководство должно рассматривать ИТ как одну из составных частей управления бизнесом в целом. Однако большинство сотрудников отделов ИТ являются специалистами в области математики, компьютерных наук, системного инжиниринга и т.п. Они не готовы к осмыслению своей деятельности через призму целей бизнеса, поскольку не обладают соответствующими навыками. Переориентация с технологии на потребителя требует психологически трудного переключения с языка традиционных технологических терминов - мипсов, байтов, пакетов и т.п. на такие, как стратегия, компетенция, деловые цели. В ответ на эту потребность сложилась специальная дисциплина - стратегическое планирование информационных технологий, которая призвана помочь "наложить" стратегию бизнеса на планирование ИТ.

В настоящее время большинство типичных ИТ подразделений привлекаются к технологическому планированию, а не к стратегическому. Такого рода планирование в современных условиях не адекватно задачам ИТ. Необходима ориентация на постоянно эволюционирующие услуги и их синхронизация с потребностями клиентов. Таким образом, стоит задача разработки стратегии ИТ как ряда целенаправленных и скоординированных действий, позволяющих использовать информационно-технологические ресурсы для создания и поддержания устойчивого конкурентного превосходства всего предприятия [1].

При использовании ИТ для создания устойчивого конкурентного преимущества предлагается следующее:

- встроить достигнутые преимущества в бизнес и продолжать работу над новыми “прорывными” решениями;
- распространить уже имеющееся превосходство на весь бизнес;
- сократить преимущества других компаний путем копирования их достижений.

В большинстве корпораций на эти задачи нацелено от 25 до 50% вложений в прикладное программное обеспечение [1].

Такое отношение к ИТ заставляет пересмотреть традиционный ответ на вопрос “В чем цель информационных технологий?” Прежний ответ - в повышении производительности, экономии финансов, подготовке более обоснованных решений - относится к способам достижения тактических краткосрочных преимуществ.

Стратегическая цель ИТ - способствовать менеджменту, реагировать на динамику рынка, создавать, поддерживать и углублять конкурентное преимущество.

2.3

Особенности перепроектированных бизнес-процессов

Из изложенного выше ясно, что перепроектированные бизнес-процессы (reengineered business process) во многом отличаются от традиционных процессов. В общем виде описать перепроектированные процессы не представляется возможным, так как они весьма разнообразны. Несмотря на это, анализ проектов по реинжинирингу, выполненных различными компаниями, позволяет выделить характерные черты типового перепроектированного процесса.

Традиционная индустриальная компания исходит из следующих посылок, восходящих к Адаму Смиту. Исполнители имеют низкую квалификацию, у них мало времени и способностей ее повысить. Из этого неизбежно вытекает, что работы (задания), поручаемые исполнителям, должны быть очень простыми. Адам Смит утверждал, что люди работают наиболее эффективно, когда они выполняют простую и легко понятную задачу. Простые задачи, однако, при их связывании с помощью иерархической структуры компании приводят к сложным процессам, что в свою очередь приводит к неэффективности, неудобству и дороговизне. Таким образом, плата за простые задания довольно высока.

Для того чтобы удовлетворять современным требованиям к качеству, сервису, гибкости и низкой стоимости, процессы должны быть простыми. Потребность в простоте процессов оказывает суще-

ственное влияние на то, как процессы проектируются и какая форма придается компании.

Рассмотрим характерные свойства, присущие перепроектированным бизнес-процессам, т.е. процессам, сформированным в ходе проведения реинжиниринга.

① Несколько работ объединяются в одну.

Наиболее характерным свойством перепроектированных процессов является отсутствие сборочного конвейера, на каждом рабочем месте которого выполняются простые задачи. Вместо этого ранее различные работы (задания) интегрируются. В описанном ранее проекте по реинжинирингу, проведенном в компании IBM Credit (см. разд. 1.2), работа нескольких специалистов, входивших в различные подразделения, объединена в работу, выполняемую одним человеком, имеющим доступ к экспертной системе с базой данных.

На практике далеко не всегда при реинжиниринге удастся сжать все шаги (задания) процесса к работе, выполняемой одним человеком. В этом случае создается команда, которая несет ответственность за данный процесс. Наличие в команде нескольких человек может приводить к некоторым задержкам и ошибкам, возникающим при передаче работы между членами команды. Однако потери здесь значительно меньше, чем при традиционной организации работ, когда исполнители процесса подчиняются различным подразделениям компании (возможно, располагающимся на различных территориях). Кроме того, при традиционной организации работ трудно (невозможно) определить ответственного за быстрое и качественное выполнение работы.

Сравнительные оценки, выполненные компаниями, которые провели реинжиниринг, показывают, что переход от традиционной организации работ к выполнению процесса одним человеком уменьшает количество людей и ускоряет выполнение процесса примерно в 10 раз (*"горизонтальное сжатие процесса"*). Уменьшается количество ошибок и отпадает необходимость держать специальную группу людей для устранения этих ошибок. Улучшается управляемость за счет уменьшения количества людей и четко распределенной ответственности между ними.

② Исполнители принимают самостоятельные решения.

В ходе реинжиниринга компании осуществляют не только горизонтальное, но и *вертикальное сжатие процессов*. Вертикальное сжатие происходит за счет того, что в тех точках процесса, где при традиционной организации работ исполнитель должен обращаться к управленческой иерархии, принимающей решения, исполнитель принимает решение самостоятельно.

При традиционной организации работ, ориентированной на выпуск массовой продукции, исходили из предположения, что исполнители не имеют ни времени, ни склонности, ни глубоких и всесторонних знаний, необходимых для принятия решений. Реинжиниринг

отбрасывает это предположение, осуществляя вертикальное сжатие, в результате которого уменьшаются временные задержки, снижается стоимость, ускоряется реакция на запросы клиента и увеличиваются полномочия исполнителей.

③ Шаги процесса выполняются в естественном порядке.

Реинжиниринг процессов освобождает от линейного упорядочения работ, свойственного традиционному подходу, позволяя выполнять работы в их естественном порядке (т.е. там, где это возможно, выполнять работы параллельно), что ускоряет выполнение процесса. Этот подход был использован при реинжиниринге процессов, выполненном компанией Kodak (см. разд. 1.2).

Делинеаризация процессов приводит к ускорению двумя путями. Во-первых, ряд работ выполняется параллельно. Во-вторых, уменьшается время, которое тратится на устранение несоответствий между предыдущими и последующими шагами процесса.

④ Процессы имеют различные варианты исполнения.

Традиционный процесс, ориентированный на производство массовой продукции, должен исполняться идентично для всех входов, производя согласованные выходы. Традиционные процессы обычно оказываются очень сложными, так как они учитывают различные исключения и частные случаи.

В современном разнообразном мире с изменяющимся рынком необходимо, чтобы процесс имел различные версии в зависимости от ситуаций, входов и состояния рынка. Новые (перепроектированные) процессы, имеющие различные версии, начинаются с некоторого проверочного шага, определяющего, какая версия процесса наиболее подходит текущей ситуации. Поэтому новые процессы в отличие от традиционных являются ясными и простыми, так как каждый вариант ориентирован только на одну соответствующую ему ситуацию (случай). Так например, в проекте IBM Credit (см. разд. 1.2) процесс имеет три версии: простые случаи (обрабатываются компьютером, без участия специалиста), средние по сложности случаи (обрабатываются специалистом с помощью экспертной системы и БД), сложные случаи (обрабатываются специалистом, привлекающим экспертов).

⑤ Работа выполняется там, где это наиболее целесообразно.

В традиционных компаниях работа организуется вокруг специалистов, сгруппированных в тематические подразделения: расчетный отдел (отделение), отдел заказов, транспортный отдел и т.п. Поэтому, если расчетному отделу требуется новый карандаш, то он обращается с заявкой в отдел заказов. Отдел заказов находит производителя, договаривается о цене, размещает заказ, осматривает товар, оплачивает его и передает в расчетный отдел. Описанный процесс является достаточно расточительным и медленным, так как он затрагивает несколько подразделений, взаимодействующих друг с другом в письменной форме. Анализ, проведенный в одной из крупных

компаний США, показал [3], что при традиционном распределении работ внутренние затраты компании на приобретение батареек стоимостью 3 дол. составили 100 дол. Кроме того, было установлено, что 35% всех заказов составляют заказы стоимостью менее 500 дол. В результате реинжиниринга расчетный и все другие отделы самостоятельно (а не через отдел заказов) заказывают себе товары стоимостью менее 500 дол.

Реинжиниринг сдвигает работу между границами подразделений, что устраняет излишнюю интеграцию и приводит к повышению эффективности процесса.

6 Уменьшение проверок и управляющих воздействий.

Проверки и управляющие воздействия непосредственно не производят материальных ценностей. Поэтому задача реинжиниринга осуществлять их только в той мере, в которой это экономически целесообразно.

Традиционные процессы насыщены проверками и управляющими шагами, не производящими ценностей для заказчика, а проверяющими соблюдение исполнителями предписанных правил. Так например, отдел заказов до исполнения заказа проверяет право клиента осуществить данный заказ, подлинность подписи клиента и финансовую состоятельность его подразделения или компании. При общей целесообразности проверок многие компании не задумываются над тем, сколько стоит проведение этих проверок. На практике довольно часто оказывается, что стоимость проверок и управляющих воздействий превосходит стоимость потерь, которые бы имели место при отсутствии проверок.

Реинжиниринг предлагает более сбалансированный подход. Вместо проверки всех выполняемых работ перепроектированный процесс часто агрегирует эти работы, делает отложенные проверки и управляющие воздействия, что заметно сокращает время и стоимость проверок.

7 Минимизация согласований.

Еще один вид работ, не производящих непосредственных ценностей для заказчика, - согласования.

Задача реинжиниринга - минимизировать согласования в ходе исполнения процесса путем сокращения внешних точек контакта.

Суть минимизации согласований можно пояснить на примере компании Ford (см. разд. 1.2). До реинжиниринга процесс оплаты счетов в компании Ford имел три точки контакта с поставщиками: в департаменте заказов в виде заказа на приобретение продукта, в отделе получения в виде документа на получение товара, в отделе оплаты счетов в виде накладной на получаемый товар. Три точки контактов приводили на практике ко многим несогласованностям. Компания Ford уменьшила количество контактов с трех до двух, что сделало ненужным согласование противоречивых документов и существенно повысило эффективность работы подразделения.

8 *“Уполномоченный” менеджер обеспечивает единую точку контакта.*

Механизм “уполномоченного” менеджера применяется в тех случаях, когда шаги процесса либо сложны, либо распределены таким образом, что интеграция их силами небольшой команды невозможна. “Уполномоченный” менеджер действует как буфер между сложным процессом и заказчиком. Менеджер ведет себя с заказчиком так, как если бы он был ответственным за весь процесс. Чтобы выполнить эту роль, менеджер должен быть способен отвечать на вопросы заказчика и решать его проблемы. Поэтому менеджер должен иметь доступ ко всем информационным системам, используемым в этом процессе, и ко всем исполнителям.

9 *Преобладает смешанный централизованный (децентрализованный) подход.*

Современные технологии дают возможность компании действовать на уровне подразделений полностью автономно (децентрализованно), сохраняя при этом возможность пользоваться централизованными данными. Например, продавцы, работающие в регионе, могут через мобильные компьютеры по беспроводной связи получать доступ к централизованной БД, находящейся в штаб-квартире компании, и использовать при составлении контрактов последние сведения по ценам, спецификациям товаров и другим условиям. Таким образом, компания может устранить бюрократические региональные структуры и повысить качество обслуживания.

Важность объединения достоинств централизации/децентрализации можно проиллюстрировать на примере работы банков. При работе с крупными корпорациями многие банки осуществляют с одним и тем же клиентом независимые финансовые отношения через свои различные подразделения. Подобный децентрализованный подход может приводить к хаосу, так как каждое подразделение отслеживает только ту часть рынка, которая соответствует его профилю. М.Хаммер [4] описывает реальную ситуацию, произошедшую с одним из банков. Для одного из своих клиентов банк установил максимальный кредит в размере 20 млн. дол. Вследствие децентрализованности этого банка несколько из его подразделений выдало данному клиенту по 20 млн. дол., т.е. клиент получил кредит в несколько раз больший, чем планировал банк, что выяснилось в связи с банкротством клиента.

Итак, чтобы избежать подобных ситуаций, банки должны иметь централизованную БД, доступную всем подразделениям, что, с одной стороны, не препятствует автономной работе подразделений, а с другой - позволяет подразделениям координировать свои действия без бюрократического аппарата.

Последствия реинжиниринга бизнес-процессов

В определении реинжиниринга подчеркивается решающая роль радикального перепроектирования бизнес-процессов. Однако необходимо помнить, что реинжиниринг начинается с перепроектирования бизнес-процессов, но не заканчивается на этом. Дело в том, что фундаментальное изменение бизнес-процессов оказывает воздействие почти на все аспекты компании.

Компания может быть представлена в виде ромба (рис. 2.1) с вершинами: бизнес-процессы; работы и структуры; системы управления и оценок; убеждения и ценности. Вершина 1 ромба соответствует бизнес-процессам компании, т.е. способу, которым работа делается. Вершина 1 определяет вершину 2, которая характеризует природу выполняемых работ и то, как люди организованы для выполнения работ. В традиционной компании процессы разбиты на простые работы, выполняемые функциональными подразделениями. В новой компании процесс разбивается на сложные (многоплановые) работы, выполняемые командами процессов.



Рис. 2.1. Компоненты бизнес-системы

Для того чтобы люди, выполняющие работу, были заинтересованы в удовлетворении потребностей клиентов в товарах или услугах, необходимы продуманные системы управления и оценок (вершина 3, на рис. 2.1), а также механизмы формирования системы ценностей и убеждений сотрудников (вершина 4) [4]. Системы управления и оценок определяют, как оценивается эффективность работы и как работа оплачивается. Ценности и убеждения сотрудников должны способствовать эффективному выполнению процессов. Например, процесс исполнения заказов клиента, спроектированный так, что он быстро и точно исполняется, не будет эффективным, если исполнители убеждены, что наиболее важны скорость и точность. Таким образом система убеждений и ценностей оказывает влияние на процессы компании (вершина 1).

Итак, для успешного функционирования компании все четыре аспекта бизнес-системы должны быть согласованы. Рассмотрим теперь более подробно последствия реинжиниринга в контексте описанных выше четырех аспектов.

① *Переход от функциональных подразделений к командам процессов.*

По сути реинжиниринг объединяет в единое целое процессы, которые, по предложению Адама Смита, много лет назад были разбиты на отдельные простые части. В традиционно организованной компании люди распределяются по отделениям, отделам, лабораториям, группам и т.п., в которых они выполняют предписанные им функции (части процессов). Эта фракционность создает множество проблем и в частности проблему несогласованности и даже противоречивости целей различных групп людей.

Реинжиниринг предлагает альтернативный подход, состоящий не в разделении людей по подразделениям, а в объединении людей в команды процессов, т.е. в группы людей, выполняющих совместно законченную часть работы - процесс. Команды процессов заменяют старые функциональные подразделения. В зависимости от сути выполняемых работ используются различные типы команд процессов. Рассмотрим наиболее часто встречающиеся типы команд.

Один тип команды объединяет некоторое число совместно работающих людей с различными специальностями, выполняющих рутинную, повторяющуюся работу. В связи с тем, что в данном случае команда выполняет повторяющуюся работу, члены команды объединяются на длительное время. Такой тип команды используется компанией Bell Atlantic [4].

Другой тип команды объединяет людей для решения некоторой эпизодической и, как правило, сложной задачи. В этом случае команда создается на время решения задачи. Команды подобного типа называют *виртуальными командами*. Такой тип команды использован фирмой Kodak при проектировании камеры (см. разд. 1.2). При завершении проекта команды этого типа расформируются, а их члены переходят в другие проекты и команды. Один человек может

быть одновременно членом нескольких виртуальных команд, распределяя свое время между несколькими проектами.

Третий тип команды использован фирмой IBM Credit (см. разд. 1.2). Этот тип команды подобен первому описанному нами типу, но состоит из одного человека.

② *Работа исполнителя изменяется от простой к многоплановой.*

Люди, работающие в команде, отмечают, что их работа значительно отличается от работы, которую они исполняли в функциональном подразделении. Член команды в отличие от сотрудника традиционного подразделения, отвечающего за отдельные задания (части процесса), несет совместно с другими членами команды ответственность за весь процесс, что требует умения не только выполнять свое задание, но и понимать весь процесс в целом и уметь при необходимости выполнять не одно, а несколько заданий. Работа члена команды становится более содержательной, так как из нее в ходе реинжиниринга устраняются излишние проверки, согласования, ожидания, вызванные преодолением границ между подразделениями традиционной компании. Члены команды фокусируют свои усилия на потребностях пользователей, а не на потребностях начальства. Пояснить характер изменений в работе можно на примере проекта компании IBM Credit (см. разд. 1.2), где после реинжиниринга работу нескольких экспертов выполняет один специалист.

③ *Требования к работникам изменяются: от контролируемого исполнения предписанных заданий к принятию самостоятельных решений.*

Традиционная ориентированная на задания компания требует, чтобы ее работники следовали предписанным правилам. Компания, завершившая реинжиниринг, требует, чтобы ее сотрудники не следовали предписанным правилам, а предлагали свои правила, т.е. члены команды уполномочены принимать самостоятельные решения. Если исполнители должны ждать указаний по их работе, то они не являются членами команды. Важность данного аспекта можно проиллюстрировать на примере проекта компании SAS (см. разд. 1.1).

Таким образом, передача полномочий исполнителям является обязательным условием проведения реинжиниринга. Проведение реинжиниринга влечет за собой изменение требований к сотрудникам, принимаемым на работу.

④ *Изменяются требования к подготовке сотрудников: от курсов обучения к образованию.*

Традиционные компании готовят своих сотрудников на обучающих курсах, цель которых обучить, как выполнять некоторую конкретную работу или как управлять той или другой специфической ситуацией. В связи с многоплановостью и изменяемостью работ, ориентированных на процессы, компании должны заботиться не только о проведении обучающих курсов, но и о непрерывном образовании своих сотрудников. Действительно при непрерывно изме-

няющемся окружении невозможно нанять людей, которые уже знают все, что от них может потребоваться.

⑤ *Изменяется оценка эффективности работы и оплата труда: от оценки деятельности к оценке результата.*

В традиционной компании схема оплаты довольно прямолинейна: людям платят за отработанное время. Понятно, что это далеко не самый эффективный способ оплаты, однако при разбиении работы на простые задания компания не имеет возможности оценить эффективность узкого задания. Кроме того увеличение эффективности узкоопределенного задания не всегда приводит к увеличению эффективности всего процесса.

После проведения реинжиниринга команда отвечает за результаты процесса, и в этом случае компания может измерить эффективность работы команды и оплатить ее в соответствии с полученным результатом. Например, в компании IBM Credit (см. разд. 1.2) работа специалиста оценивается не по количеству отчетов, а по числу и выгоде, полученной от заключенных им договоров.

Реинжиниринг приводит к тому, что компании пересматривают базовые предположения об оплате труда, свойственные традиционному подходу:

- Эффективность работы сотрудника в текущем году не является гарантией его эффективной работы в следующем году. По этой причине базовая зарплата сотрудника меняется мало, награду за высокую эффективность он получает в виде премий.

- Жалование сотрудника определяется не столько временем, проведенным на работе, важностью выполняемой работы, трудовым стажем, количеством подчиненных, занимаемой должностью, сколько эффективностью его работы, оцениваемой по конечному результату.

⑥ *Критерий продвижения в должности изменяется: от эффективности выполнения работы к способности (умению) выполнять работу.*

Одним из последствий реинжиниринга является проведение четкого различия между продвижением сотрудника и эффективностью его работы. Наградой за эффективность работы должна быть премия, а не продвижение по службе. Продвижение по службе есть функция от способностей сотрудника, а не от эффективности его работы, т.е. принцип компаний должен быть таков: "платим за эффективность, продвигаем за способности". Несмотря на очевидность этого принципа, он часто нарушается. Типичным является такое рассуждение: если N хороший программист, то он подходит на должность руководителя лаборатории программистов. Подобный вывод часто оказывается ошибочным, и компания получает плохого руководителя за цену хорошего программиста.

7 *Изменяется цель исполнителя: от удовлетворения потребностей начальника к удовлетворению потребностей клиентов.*

Реинжиниринг вызывает существенный сдвиг в культуре компании. Реинжиниринг требует от исполнителей убежденности, что они работают для клиентов, а не для своих начальников (боссов). Исполнители будут верить в это в той степени, в которой практика работы компании подтверждает это. Так например, Херох Corporation не только говорит своим сотрудникам, что зарплату им платят клиенты, но и реализует это высказывание следующим образом: основная часть премии менеджеров зависит от степени удовлетворения ими клиентов.

8 *Функции менеджеров изменяются от контролирующих к тренерским.*

В результате реинжиниринга бизнес-процессы становятся проще, а отдельные задания (шаги) процесса, выполняемые исполнителем, становятся сложнее. Так, в проекте IBM Credit (см. разд. 1.2) работу нескольких человек выполняет один специалист. Усложнение работ, выполняемых исполнителями, приводит к тому, что уменьшается работа менеджеров по контролю за ходом выполнения процесса. В связи с тем, что команда процесса (см. подразд. 4.8.3) полностью отвечает за выполнение своего процесса, устраняются управляющие воздействия на исполнителей со стороны менеджеров. Функции менеджеров изменяются, их задача состоит теперь не в выдаче управляющих и контролирующих воздействий, а в помощи членам команды решать проблемы, возникающие у них в ходе выполнения процесса. Таким образом, менеджер выполняет функции тренера, который непосредственно не участвует в работе команды, но помогает команде выполнить ее работу с минимальными непроизводительными затратами. Именно этот вид деятельности требует от менеджера подлинного профессионализма.

Традиционная практика недооценивает как работу исполнителя, так и работу менеджера. Недооценка роли исполнителя выражается в том, что для него вершина успеха состоит в переходе из исполнителей в менеджеры, т.е. фактически традиционные компании оценивают управленческую деятельность выше, чем деятельность по производству товаров и услуг. Недооценка роли менеджера состоит в том, что утверждается: любой хороший исполнитель может стать хорошим менеджером. Ошибочность этого утверждения очевидна любому спортивному болельщику, знающему, что не любой, даже выдающийся спортсмен, может стать хорошим тренером.

9 *Организационная структура компании изменяется от иерархической (многоуровневой) к более "плоской".*

В традиционной компании организационная структура играет важную роль, так как она является механизмом, с помощью которого решаются основные проблемы компании. Действительно, в тра-

диционной организации основной единицей является функциональное подразделение - совокупность людей, объединенных по подобию выполняемых ими задач (заданий). При этом компания как целое состоит из функциональных подразделений, организованных тем или иным способом. В функциональных компаниях все связанные функциональные отделы объединяются в единое функциональное отделение. Например, все отделы продаж объединяются в отделение продаж. Возможно объединение отделов по территориальному принципу, например, Западное отделение компании.

Организационные структуры устанавливают границы взаимодействия между подразделениями и определяют иерархию принятия решений. Таким образом, процесс разбивается на отдельные части, выполняемые в различных подразделениях. При этом работа менеджеров в значительной степени состоит в контроле за исполнителями и в "склеивании" отдельных работ в единый процесс.

После проведения реинжиниринга значительно сокращается работа, выполняемая менеджерами (т.е. уменьшается требуемое количество менеджеров), и меняется ее характер (от контролирующей к тренерской). Действительно менеджер, осуществляющий контролирующие функции, обычно не может работать более чем с семью подчиненными менеджерами или исполнителями. Менеджер, осуществляющий тренерские функции, может работать примерно с тридцатью людьми. Изменение соотношения от 1 к 7 на 1 к 30 приводит к тому, что значительно сокращается количество управляющих уровней в иерархической структуре, в связи с чем важность организационной структуры уменьшается.

10 *Административные функции изменяются от секретарских к лидирующим.*

В традиционной компании администрация оторвана от непосредственных исполнителей и клиентов, она выполняет функции секретаря, а не руководителя. Одним из последствий реинжиниринга является изменение роли руководящей администрации. Уменьшение количества управляющих уровней в иерархической структуре приближает администрацию к непосредственным исполнителям и к клиентам. В перепроектированной компании успешное выполнение работы в основном зависит от членов команды, а не от функциональных менеджеров. Следовательно, администрация должна исполнять функции лидера, способствующего словом и делом укреплению убеждений и ценностей исполнителей.

Администрация несет общую ответственность за перепроектированный процесс, но не имеет непосредственного воздействия на людей, выполняющих этот процесс, так как члены команды и их тренер работают довольно автономно. Администрация осуществляет влияние на эффективное исполнение процессов за счет того, что в перепроектированных процессах с помощью систем управления обеспечивается мотивация членов команды.

Структура традиционной и новой компании

Существует много различных способов спроектировать современную компанию, и есть много различных способов реализации одного и того же проекта. Тем не менее, можно высказать некоторые общие соображения о новой компании.

С одной стороны, в новой компании будут в основном работать те же люди, что и в старой. Только небольшое количество энергичных и опытных специалистов будет нанято, а некоторые специалисты уволены, так как их знания недостаточны или они не смогли приспособиться к изменению своих обязанностей.

С другой стороны, компания уже не та, что была раньше. Теперь люди решают другие задачи и на ином качественном уровне. Им предоставлены новые способы решения задач, люди вовлечены в работы, требующие большей ответственности. Исполнители ощущают свою большую значимость.

Клиенты обнаруживают, что теперь они обрели собственное лицо в глазах компании и стали значительно более важными для нее, чем были раньше. Им становится приятнее иметь дело с новой компанией. Звучит слишком хорошо? Но что произошло? Разумеется, нечто революционное.

Традиционная иерархическая функциональная компания имеет менеджеров нескольких уровней (например, верхний, средний, линейный). В основании иерархии, там, где выполняется почти вся работа, находятся исполнители, обслуживающие клиентов. Клиенты оказываются практически “на дне” структуры (рис. 2.2). В новой компании клиенты находятся не внизу, а сбоку от компании, как если бы они были равноправными партнерами; непосредственные исполнители, взаимодействующие с клиентами, являются операторами процессов; менеджеры среднего уровня являются руководителями групп, отвечающих за соответствующие процессы (руководители процессов).

В компании, основанной на процессах, можно выделить несколько типовых ролей сотрудников. Каждый конкретный человек может выступать в одной роли или сразу в нескольких (более подробно см. разд. 2.6).

Президент компании назначает владельцев ресурсов и владельцев процессов: по одному для каждой функции и для каждого процесса в компании (в большой компании между президентом и владельцами ресурсов и процессов могут стоять должностные лица, отвечающие за различные сферы бизнеса).

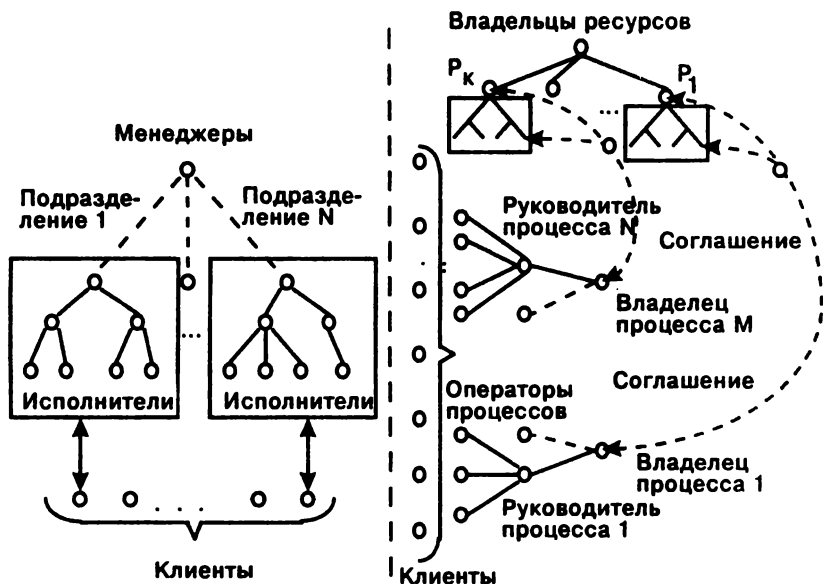


Рис. 2.2. Структура старой (слева) и новой (справа) компании

Владелец процесса несет оперативную ответственность за ресурсы, предоставленные в его распоряжение, чтобы выполнить некоторый конкретный процесс. Владелец ресурса имеет долговременную ответственность за ресурсы, относящиеся к его конкретной функции.

Владелец процесса заключает соглашение с операторами процесса, которых он приглашает к себе на работу. Заключается трехстороннее соглашение между оператором, владельцем процесса и владельцем ресурса. Предложения по такому соглашению представляются оператору и, если тот соглашается с ними, все стороны подписывают и принимают их. Кроме того, владелец процесса назначает руководителя (лидера) для каждого из своих конкретных процессов. Этот лидер несет оперативную ответственность за порученный ему конкретный процесс.

Различие между старой и новой компанией проявляется в основном в работе над проектом (в нашей терминологии - процессом). В старой компании руководитель проекта в общем случае сталкивается с тем, что работа должна выполняться в нескольких функциональных подразделениях. Когда руководитель проекта хочет нанять исполнителей у линейных менеджеров, он обычно встречает недружелюбный прием. Линейные менеджеры имеют несколько обязанно-

стей. Они должны обеспечить, чтобы в их подразделении работы выполнялись правильно и соответствовали функциям подразделения. Они должны также заботиться о своем персонале и о повышении его квалификации. Линейные менеджеры понимают, что руководители проектов необходимы и выполняют важную работу, но они не хотят терять свои ресурсы. Переговоры между руководителями проектов и линейными менеджерами тянутся без конца. Всякий, кто работал над проектом, знает, что необходимо отчитываться и перед руководителем проекта, и перед линейным руководителем, которые не всегда согласны друг с другом. Ситуация является конфликтной с самого начала.

В новой компании руководитель процесса (см. рис. 2.2) также набирает сотрудников из различных подразделений, исходя из их опыта и профессиональной подготовки, а также из тех заданий, которые должны выполняться процессом, и организует их в команду (бригаду). Члены команды - это операторы процесса, каждому из них отводится свой участок работы в соответствии с его опытом и навыками. Работа команды состоит из некоторого числа действий (заданий), связанных вместе.

Работа выполняется, исходя из задач, поставленных конкретным клиентом, и заканчивается созданием некоторых потребительских ценностей для этого клиента. В соответствии с современной терминологией работа - это "проект" (например, процесс разработки продукта: от требований на продукт к реальной продукции). Процесс имеет начало и конец, и когда он завершается, использованные в нем ресурсы могут быть задействованы для выполнения другой работы. Возможно, в одно и то же время будет выполняться несколько конкретных реализаций (экземпляров) одного вида процесса. Каждая реализация имеет своего руководителя и своих операторов. Руководители процесса подотчетны владельцу процесса. Последний определяет процесс: устанавливает и отслеживает его цели, условия выполнения и т. д., в то время как руководитель процесса выполняет его конкретные реализации.

Может показаться, что нет существенной разницы между старой линейной компанией и компанией, ориентированной на процессы. Однако разница есть, и она носит решающий характер.

В первую очередь для новой компании процессами являются не только проекты, как для старой, но и все виды работ вообще, в том числе рутинные и повторяющиеся работы. Примером циклически повторяющейся работы является оформление продаж: от заявки до заказа. Эта повторяющаяся работа может выполняться одним сотрудником. Когда он заканчивает работу с одним клиентом, он переходит к следующему, и т.д. Компания может по-разному координировать распределение работы между сотрудниками, занимающимися этим видом работы. Например, каждая область может обслу-

живаться отдельным сотрудником или координатор продаж может участвовать во всех конкретных реализациях процесса.

Другим примером циклически повторяющегося процесса является авиарейс: от регистрации до выдачи багажа. Здесь одна летная команда, подкреплённая наземными службами, начинает обслуживание нового рейса сразу же по завершении работы по предыдущему. В линейных компаниях такие работы явно не выделяются. Это не проекты, хотя их выполнение “пронизывает” несколько различных подразделений, и они делаются как бы вопреки линейной структуре организации. В новых компаниях эти работы “поднимаются на поверхность” и становятся видимыми для всех, получают статус и признаются тем, чем они, собственно говоря, и являются на самом деле, - процессами.

Другое существенное различие между линейными (функциональными) и процессоориентированными компаниями состоит в отсутствии (или в резком сокращении числа) в компании нового типа менеджеров среднего звена, которые несут ответственность за то, что и каким образом делается в их функциональной области. В новой компании менеджеры в качестве надзирателей больше не нужны. Здесь ответственность перед высшим руководством (которое не показано на рис. 2.2) несёт руководитель процесса. Он же договаривается с руководством о стоимости своего процесса. Активы (деньги), которые предоставляются руководством, должны использоваться руководителем процесса для привлечения ресурсов, будь то люди, оборудование или программное обеспечение, как внутри, так и вне компании. Высшее руководство отвечает за выполнение работы и за координацию различных процессов на уровне интерфейсов между этими процессами.

Компания, ориентированная на процессы, группирует своих служащих в соответствии с их областью компетентности. Для каждой из этих областей назначается владелец ресурса, который имеет исполнителей, но не имеет денег; владелец некоторого процесса имеет деньги, но не имеет исполнителей. Одна из основных задач владельца ресурса - это ведение переговоров с владельцами процессов о “продаже” труда своих исполнителей для того, чтобы получать деньги на покрытие своих расходов (зарплата и т.п.). При такой организации работы владельцы ресурсов заинтересованы в том, чтобы предлагать владельцам процессов именно тех исполнителей, которые им нужны. Владелец ресурса, как и владелец процесса, подотчётен непосредственно высшему руководству компании.

В новой компании служащие склонны смотреть на свою работу более позитивно, чем прежде. Все вместе они стараются удовлетворить потребности клиентов компании. От них ожидается выполнение того, что непосредственно требуется в конкретном исполняемом процессе, а не то, что диктуется их функциональной ролью, поэтому

их задания становятся более интересными и разнообразными. Сотрудники не ощущают того надзора, который был раньше, они осознают рост своих полномочий и даже начинают чувствовать себя предпринимателями. Поскольку становится легче измерять степень удовлетворения клиента конкретной работой, упрощается выплата премий исполнителям этой работы. Перспективы служебного роста расширяются и делаются более определенными, поскольку напрямую зависят от индивидуальных способностей. Роль руководителей процессов и владельцев ресурсов очень важна, поэтому поиску этих сотрудников уделяется особое внимание. В линейной компании, наоборот, нередко хороших технических специалистов поощряют продвижением в менеджеры. В результате можно получить плохого менеджера вместо хорошего специалиста. В компании, ориентированной на процессы, не придется совершать такие ошибки. Здесь всегда ценятся знающие компетентные сотрудники, и компания имеет различные возможности их поощрения.

Дж.Карлсон [2] рассказывает следующую историю. Два каменотеса были поставлены изготавливать прямоугольные блоки из гранита. Когда их спросили, чем они занимаются, один из них ответил со скукой в голосе: “Я вытесываю из этих камней прямоугольные блоки”, - а другой гордо заявил: “Я участвую в строительстве собора”. В реконструированной компании каждый сотрудник имеет четкое представление о цели своей команды и о способах ее достижения. Каждый знает, как будет измерена успешность его работы. Каждый понимает и уважает те ценности, которые привносятся в команду и бизнес в целом его коллегами, и каждый осознает свое место не только в команде, но и в компании.

2.6

Участники проекта по реинжинирингу и их роли

М. Хаммер и Дж.Чампи [4] для среднего проекта выделяют следующих участников реинжиниринга компании:

- лидер проекта - член высшего руководства компании, который возглавляет организацию и проведение реинжиниринга;
- владельцы процессов - менеджеры, отвечающие за обновляемые процессы;
- руководящий комитет (совет) наблюдателей (steering committee) - комитет, образованный из представителей высшего руководства компании, основная цель которого - определение общей стратегии по реинжинирингу и контроль выполнения работ по проекту;

- “царь” (reenginiring czar) - специалист компании, отвечающий за развитие методик и инструментариев поддержки реинжиниринга, а также координирующий выполнение различных проектов в рамках этой компании;

- команда по реинжинирингу - группа специалистов (сотрудники компании, а также эксперты и разработчики, приглашенные со стороны) для проведения реинжиниринга выбранного процесса (процессов).

Лидер проекта назначает владельцев процессов и ресурсов, которые привлекают команду по реинжинирингу для обновления этих процессов. Разработка осуществляется с участием “царя” и под контролем руководящего комитета. Комитет возглавляется лидером проекта по реинжинирингу.

Лидер проекта - это руководитель, инициирующий проведение реинжиниринга в своей компании и берущий на себя основную ответственность и риск, связанные с проектом. Его основная задача - формирование представления о будущей обновленной компании и обеспечение должной мотивации остальных сотрудников.

Лидера проекта невозможно назначить. Он сам “выдвигает себя” на эту роль и служит источником энергии и энтузиазма для всех остальных участников реинжиниринга. Лидер создает в компании соответствующую атмосферу, в которой каждый сотрудник сможет проявить готовность отказаться от старых правил работы.

Лидер проекта должен иметь достаточные полномочия для выполнения перечисленных задач. Помимо этого очень важно, чтобы его личные качества соответствовали взятым на себя обязательствам: без сильного, агрессивного и высокопрофессионального руководства довольно трудно (если вообще возможно) преодолеть сопротивление среднего звена (в первую очередь, функциональных подразделений), которое будет затронуто изменениями в наибольшей степени.

Владелец процесса - это руководитель высшего звена, который во время реинжиниринга несет ответственность за обновляемый процесс. Если в компании еще не выделены процессы, то ответственность за них распределяется между функциональными подразделениями. Задача лидера проекта в этом случае состоит в идентификации основных процессов компании и назначении владельцев для них.

Владелец процесса сам не выполняет реинжиниринг. Его задача состоит в привлечении квалифицированной команды процесса и обеспечении ей нормальных условий для работы над проектом. Владелец процесса участвует в проекте как наблюдатель и оппонент. Он же отвечает за должную мотивацию членов команды.

Руководящий комитет не является обязательным участником работ по реинжинирингу, поскольку его функции может выполнять лидер проекта. Однако в больших компаниях, выполняющих одновременно несколько проектов, целесообразно наличие такого коми-

тета. Как правило, в него входят владельцы процессов, которые планируют общую стратегию компании по реинжинирингу. В частности определяются приоритеты проектов, выполняемых в компании, а также решаются конкретные проблемы, требующие совместных усилий владельцев различных процессов (например, преодоление конфликтных ситуаций).

“Царь” выполняет функции оперативного руководителя всех работ по реинжинирингу в компании. Как было показано, в этих работах принимает участие довольно большое число сотрудников. Поскольку лидер не имеет возможности осуществлять повседневное оперативное управление этими сотрудниками, он нуждается в помощнике, берущем на себя руководство штатом по реинжинирингу. Этот помощник, которого называют также “царем”, подчиняется непосредственно лидеру проекта, он выполняет две основные функции: во-первых, обеспечивает работу по каждому конкретному проекту; во-вторых, координирует работы по всем одновременно выполняемым проектам. “Царь” находится в курсе современных методов проведения реинжиниринга и соответствующих инструментариев. Он вводит в курс дела вновь назначаемых владельцев процессов, а также помогает им привлекать специалистов со стороны для формирования команды по реинжинирингу. Наконец, “царь” организует оперативное взаимодействие владельцев отдельных процессов в тех случаях, когда необходима координация разработок, выполняемых в рамках их проектов.

Команда по реинжинирингу - это коллектив сотрудников, непосредственно выполняющих проект. Назначение членов команды по реинжинирингу на выполнение некоторого проекта сводится к распределению работ по исполнителям. Помимо сотрудников, которые непосредственно заняты моделированием бизнеса (см. выше), для выполнения проекта могут привлекаться следующие ресурсы и группы [5]:

- **Эксперт по методу** - это специалист или группа специалистов, отвечающие за используемую методологию. Они должны быть экспертами по данному методу и оказывать команде помощь, необходимую для его применения в данном конкретном проекте. Эти специалисты должны разбираться и в деятельности компании, что требует от них знания соответствующей отрасли, текущей организации компании, структуры ее продукции, а кроме того - понимания степени компетентности остальных членов команды по реинжинирингу.

- **Группа обеспечения качества** - сотрудники, ответственные за то, чтобы бизнес и система разработки бизнеса имели высокое качество. В частности, они должны гарантировать надежность структуры и достаточность документации для новой организации бизнеса. Для обеспечения качества очень полезно проведение обсуждений проекта (способы организации обсуждений будут рассмотрены в подразд. 4.8.3).

- *Группа документирования.* Документирование моделей бизнеса должно выполняться членами команды по реинжинирингу. Описание потоков работ, понятное менеджерам и служащим компании, должны делать люди, обладающие специальными навыками. Эта группа планирует обучение новым способам ведения бизнеса. Обучение должны проходить менеджеры и лидеры процессов, операторы, клиенты и т.д.

- *Координатор (или группа координации)* отвечает за повторное использование моделей смоделированной инфраструктуры. В больших компаниях или при наличии в компании нескольких филиалов часто возникает необходимость использования одной и той же инфраструктуры в нескольких местах. Координатор должен оценить, в какой степени проект допускает повторное использование того, что уже есть, а также потенциальные возможности повторного применения разработанных моделей. Координатор должен поддерживать тесную связь с командой по реинжинирингу. Хотя повторное использование может не дать выигрыша для какого-либо конкретного проекта, отдача от проведенной работы может проявиться при разработке последующих проектов. Таким образом, не следует списывать затраты на координацию на один проект: эти затраты распределяются между несколькими последовательно разрабатываемыми проектами. Таким образом, координация и управление библиотекой повторного использования должны иметь межпроектный статус.

- *Группа прототипирования* исследует различные решения на ранней стадии разработки. Работы по прототипированию рекомендуется проводить совместно с заказчиками в начале работы над проектом. Следует подчеркнуть, что при использовании современных интегрированных инструментальных средств (например, ReThink, см. разд. 9.3), основывающихся на технологии экспертных систем, прототип (прототипы) возникают как естественный результат работы над проектом, и отдельная группа прототипирования не нужна.

- *Лидеры процессов и владельцы процессов* отвечают за создание поддерживающей информационной системы. Они работают в тесном контакте с разработчиками бизнеса и с разработчиками поддерживающей ИС.

- *Штат менеджера проекта.* В этот штат может быть включен, например, администратор проекта, который отвечает за выполнение текущих планов с учетом стоимости и сроков разработки. В больших проектах такой администратор просто необходим.

Роли и обязанности

Перейдем к описанию штатов, которые необходимы при функционировании реконструированного бизнеса, основанного на процессах. Ранее мы уже касались этой темы. В этом разделе дадим более полную характеристику различных ролей (обязанностей и ответ-

ственности), которые играют сотрудники в компаниях подобного типа.

В компании, базирующейся на процессах, владелец процесса “закупает” внутри компании ресурсы и услуги у владельцев ресурсов. Этот подход не может быть реализован при выполнении первого проекта по реинжинирингу, поскольку новая компания прежде чем сможет работать эффективно, обычно нуждается в “настройке”, необходимой для построения внутренней системы поддержки и обеспечения процессов. Но спустя некоторое время перестроенный бизнес станет достаточно зрелым для перехода к настоящей динамичной организации. Владельцы процессов при этом заинтересованы в хорошей репутации своих процессов, дающей им право приглашать к себе лучших сотрудников. Каждый сотрудник будет стараться выполнять свои функции наилучшим образом, чтобы у владельцев процессов было желание “покупать” его услуги.

В бизнесе, основанном на процессах, можно выделить несколько типовых ролей сотрудников. Каждый конкретный человек может выступать в одной роли или сразу в нескольких.

Перечислим обязанности основных категорий сотрудников компании, которые являются характерными для компаний, основанных на процессах [5], а затем рассмотрим различные виды соглашений, регулирующих обязательства и условия выполнения работ.

Президент:

- ставит оперативные и долгосрочные цели;
- определяет стратегии бизнеса;
- осуществляет общий контроль за бюджетом и финансовой деятельностью;
- обеспечивает развитие бизнеса и организационной структуры компании;
- назначает владельцев процессов и владельцев ресурсов;
- контролирует деятельность владельцев процессов и владельцев ресурсов.

Владелец ресурса:

- распределяет операторов процессов или другие ресурсы на различные бизнес-процессы;
- разрешает конфликты, возникающие при распределении ресурсов;
- заключает соглашения с сотрудниками (операторами), из которых они получают ясное представление о своей долгосрочной роли в разработках компании. Владелец ресурса обеспечивает каждого сотрудника индивидуальным планом работ. Эти планы должны обсуждаться и пересматриваться ежегодно;
- обеспечивает повышение квалификации своего персонала и ведет проверку его компетентности;

- составляет бюджет долгосрочного обучения и стратегической подготовки (не связанной специально с каким-либо конкретным процессом);

- принимает на работу (совместно с владельцем процессов) операторов процессов;

- поддерживает служащих в переговорах, профессиональном росте, при разрешении конфликтов по использованию ресурсов и т.п.

Владелец процесса:

- назначается для каждого бизнес-процесса в компании. Непосредственные задачи этого сотрудника определяются характером того процесса, за который он отвечает;

- разрабатывает процесс и обеспечивает, чтобы он соответствовал стратегическим бизнес-планам компании;

- определяет интерфейс процесса на основе анализа объектов, участвующих в данном процессе. Если какие-либо из этих объектов участвуют и в других процессах, то владелец процесса определяет, как эти объекты должны отрабатывать свои обязательства в различных процессах. Эта работа обычно выполняется совместно с владельцами других процессов;

- определяет цели процесса и обеспечивает их достижение;

- планирует бюджет для всех экземпляров процесса и, следовательно, общий бюджет процесса;

- назначает лидера (лидеров) экземпляров процесса;

- распределяет ресурсы для всех экземпляров процесса и заключает соглашения с каждым оператором процесса;

- оперативно дорабатывает процесс в случае необходимости; для проведения этого вида работ в бюджете выделяются соответствующие средства;

- участвует в долгосрочном планировании потребностей в ресурсах (основным ответственным за этот вид деятельности является владелец ресурсов);

- обеспечивает развитие процесса и улучшение его качества.

Лидер процесса:

- несет ответственность за конкретный экземпляр процесса, при этом он имеет много общих черт с лидером проекта. В небольших компаниях эти обязанности нередко выполняет одно и то же лицо. Лидер процесса в случае необходимости может вносить поправки в свой экземпляр процесса, например, если он работает в филиале предприятия;

- обеспечивает эффективное выполнение конкретного экземпляра процесса в соответствии с бюджетом;

- согласует возможные поправки или изменения в конкретном процессе с владельцем процесса;

- осуществляет управление и оперативное планирование ресурсами (операторами, привлеченными на контрактной основе);

- выявляет и разрешает конфликты (совместно с операторами процесса, владельцем процесса и владельцем ресурсов);
- подписывает соглашения с привлекаемыми операторами процесса.

Оператор процесса.

Все сотрудники компании в конечном итоге являются операторами процессов. Некоторые из них назначаются президентом и владельцами процессов. Такие операторы играют в компании особые роли, которые рассмотрены выше. В остальных случаях операторы процессов выполняют следующие задания:

- находят для себя работы и заключают договора в одном или нескольких бизнес-процессах в компании;
- требуют заключения и выполнения всех соглашений, необходимых для успешного выполнения работы;
- составляют совместно с лидерами процессов подробные индивидуальные планы со сроками выполнения работ;
- выполняют работы в конкретных процессах;
- следят за своим профессиональным ростом.

Виды соглашений

Обязательства и профессиональный рост каждого сотрудника в компании могут регулироваться различными соглашениями, в том числе следующими:

- ① Личный план профессионального роста, описывающий основные курсы, которые должен закончить оператор процесса, а также его возможные продвижения в компании на ближайшие один - три года.
- ② План участия в процессах, указывающий, какой процент времени оператор затрачивает на участие в различных процессах в течение года.
- ③ Соглашения, определяющие условия, на которых оператор процесса обязуется выполнять некоторую работу. Этот тип соглашений используется по преимуществу для консультантов, приглашаемых со стороны и являющихся временными участниками проекта.

Поскольку личный план профессионального роста и соглашение о назначении не являются специфичными для проектов по реинжинирингу, рассмотрим здесь только план участия в процессах.

План участия в процессах

Время действия этого соглашения ограничивается обычно одним годом, и его пересмотр совпадает с пересмотром сотрудниками компании своих решений об участии в тех или иных процессах. В большинстве случаев переговоры оператора с владельцем процесса берет на себя владелец ресурса. В итоге оператор, владелец ресурса и владелец процесса подписывают соглашение. В нем должно отражаться распределение рабочего времени оператора по различным процес-

сам. Если в момент принятия соглашения это не было возможно, то необходимые дополнения к соглашению должны быть введены позднее, при первой возможности. Если есть намерение заключить соглашение на срок, более длительный, чем один год, то это должно быть явно отмечено в тексте соглашения. Помимо всего прочего, соглашение содержит информацию о том, как должна происходить его отмена. Форма типичного плана участия в процессах содержит: название процесса; процент времени, которое данный оператор затрачивает на него; задачи, которые он решает, и имя владельца процесса.

ЛИТЕРАТУРА

1. Boar B. H. **The art of strategic planning for information technology**//Crafting Strategy for the 90s. - John Wiley&Sons, 1993.
2. Carlzon J. **Moments of Truth**. - MA: Ballinger Publishing Company. 1987.
3. Davenport T.H. **Business Innovation, Reengineering Work through Information Technology**. - Boston: Harvard Business School Press, 1993.
4. Hammer M. and Champy J. **Reengineering the Corporation: A Manifesto for Business Revolution**. - N-Y: HarperCollins, 1993.
5. Jacobson I., Ericsson M., Jacobson A. **The Object Advantage: Business Process Reengineering with Object Technology** //ACM Press. - Addison-Wesley Publishing, N-Y: 1995.
6. Johansson H., McHugh P., Pendlebury J. and Weeler III W. **Business Process Reengineering. Breeakpoint Strategies for Market Dominance**. Chichester: John Wiley & Sons, 1993
7. Manganelli R. **Merging BPR and Strategy Implementation**. The National Publication for BPR Enterprise Reengineering., V. II-Jssue N6. - September 1995.
8. Willoch B.E. **Business Process Engineering. Vinnende arbeidsprosesse og organisasjonsstrukturere i forandringens decennium**. - Norway: Fagbogförlaget, 1994.

3

МОДЕЛИРОВАНИЕ БИЗНЕСА

3.1

Модель бизнеса

Каждая компания представляет собой сложный организм, изменяющийся во времени. В каждый данный момент картина, которую представляет компания, является лишь малой частью правды о ней - большая часть скрыта в сложных деталях. Полностью понять сущность компании невозможно. К счастью, этого и не требуется. Однако желательно исследовать компанию настолько глубоко и детально, чтобы каждый, имеющий с ней дело, в любой ситуации располагал достаточной информацией для принятия наилучших из возможных решений. Разумеется, даже достаточное количество информации не гарантирует успеха, хотя знание безусловно уменьшает риск. В наиболее преуспевающих компаниях некоторые их сотрудники имеют уникальные знания о компании в целом: о ее стратегиях, целях, процессах, продуктах, услугах, ресурсах и об отношениях между перечисленными факторами. Задача заключается в том, чтобы передавать эти знания тем людям, которые должны быть осведомлены о важных аспектах деятельности компании. Известно, что сложные вещи проще понимать, если они тем или иным образом зрительно представлены, а не только описаны словами. Формальный способ визуализировать нечто - это построить модель. Ниже будет введено понятие *модели* и рассмотрены типы моделей, удобные для тех или иных целей. Будет показано также, как все эти модели могут быть связаны вместе, чтобы получить полную картину компании. Следует помнить, что модель не адекватна моделируемой сущности - это всего лишь более или менее верное представление этой сущности.

Практически в любой области люди используют тот или иной вид моделей (математических, физических или компьютерных), чтобы иметь более ясное понятие о том, что они делают. Существует два

основных способа описания моделей - статическое и динамическое. *Статическое описание* рассматривает структуру модели, т.е. такие ее аспекты, в которых можно пренебречь временем. *Динамическое описание* рассматривает поток событий, т.е. изменение моделируемых явлений во времени, которым нельзя пренебречь с точки зрения задач, решаемых компанией. Таким образом, представляется совершенно естественным использовать различные модели для описания разных аспектов компании. Очевидно, что рассматривать в одной модели все аспекты не только не нужно (поскольку это неэффективно), но и невозможно.

Действительно деятельность компании можно рассматривать с точки зрения различных людей: оператора процесса, лидера процесса, исполнительного директора, заказчика, акционера, партнера компании, продавца продукции компании и т.д. С точки зрения каждой из перечисленных выше категорий людей компания выглядит по-разному, т.е. каждой категории необходимы различные модели. Так, исполнительный директор должен иметь общую картину, включающую все аспекты компании в целом: концепцию бизнеса, процессы, продукцию, персонал, инвестиции, финансы, перспективы и т.д. Эта картина должна быть в высшей степени интегрированной. Для того чтобы управляющий персонал мог принимать правильные решения в любых ситуациях, необходимо иметь набор моделей, которые ясно описывают различные аспекты компании и их взаимоотношения. В моделях, используемых на верхнем уровне управления, самое главное - это краткость и понятность: здесь подчеркнуты основные сущности, а детали скрыты. Каждой категории людей, работающих с компанией, требуется в точности та информация, которая им необходима для их деятельности. Впрочем, это утверждение не следует понимать слишком буквально. Очевидно, что в ряде случаев члену команды целесообразно знать больше о том, что непосредственно его касается, и эту информацию ему следует предоставить.

Несколько участников могут иметь потребность в доступе к одной и той же модели. Например, исполнительный директор должен знать, что сообщено акционерам. Для руководителя (лидеров) процессов и руководства компании важно иметь одинаковую общую картину того, чем занимается компания, чтобы различные группы могли говорить на общем языке. Наконец, что совершенно очевидно, различные модели должны согласоваться. Картина, представленная исполнительному директору об экономике и финансах компании, должна согласовываться с тем, что он видит (видел) в действительности.

Обычно компания объединяет различные категории взаимодействующих с ней людей, и теоретически можно разработать модель для каждого из них. На практике, к сожалению, дело обстоит не совсем так. Акционеры получают одну модель описания компании, персонал по продажам - другую, клиенты - третью. Эти модели

обычно не вполне согласуются друг с другом, так как их разработку в большинстве компаний никто не координирует. Но чтобы полнее использовать потенциал компании, следует сосредоточить внимание на наиболее сложных для понимания аспектах, которые требуют обучения, уточнения, улучшения, изменения.

Одна из наиболее важных моделей - модель бизнеса, с помощью которой проясняются функции компании во внешнем мире.

Модель бизнеса показывает, что является *окружающей средой компании* и как компания взаимодействует с этой средой. Под *окружающей средой* понимают все, с чем компания взаимодействует в ходе выполнения своих бизнес-процессов - в частности, клиентов, партнеров, субподрядчиков и т.д. Модель бизнеса показывает работникам всех уровней, что должно быть сделано, когда и как именно. В общем случае необходима не одна, а несколько интегрированных и согласованных бизнес-моделей.

Ключевой элемент модели бизнеса - описание архитектуры компании, т.е. описание ее наиболее важных статистических структур: отделений, отделов и др. Однако просто организационная схема плохо отражает архитектуру компании. Другие важные статические структуры - процессы (их описание, но не исполнение), продукция, человеческие и технические ресурсы. Структуры состоят из взаимосвязанных элементов. Элементы имеют ответственных за них владельцев - кого-либо из сотрудников компании. Элементы осязаемы: они имеют содержание, им может быть присвоено значение (иногда несколько значений), у них есть ограничения. Обычно динамику - поток событий в компании - не рассматривают как часть архитектуры. Определяя архитектуру, как правило, не принимают в расчет ни совместное функционирование элементов, ни то, что они делают в данной ситуации или как они взаимодействуют, чтобы выполнить свое назначение. Наличие некоторого потока событий (например, процесса) имеет отношение к архитектуре, но сам способ протекания событий не относится к архитектуре. Действия и принимаемые решения, образующие поток событий, являются деталями отдельного процесса. Следовательно, во многих случаях важно описать динамику бизнеса и включить ее в модель, но, как правило, динамика не учитывается в архитектуре модели.

Для удобства работы с моделью бизнеса необходимо ограничить представляемую ею информацию. **Описывая бизнес** целой компании, целесообразно исключать массу деталей. Модель должна представлять в точности то, что хотят показать, проиллюстрировать, объяснить, понять, обсудить или улучшить, - ничего больше, но и не меньше. Обычно модель бизнеса разрабатывается только для тех частей компании, которые выполняют ключевые бизнес-процессы. Ключевые бизнес-процессы - это те, в которых участвуют клиенты, и те, при помощи которых компания зарабатывает деньги.

Итак, модель бизнеса показывает функцию компании во внешнем мире: что она делает, как и когда. Модель должна представлять архитектуру, т. е. статические структуры компании, а кроме того, - различные потоки событий, т. е. динамическое поведение элементов архитектуры. Техника моделирования должна быть достаточно мощной для построения как общих моделей компании в целом, так и ее детальных описаний.

Чтобы дать представление о том, какой должна быть модель бизнеса, рассмотрим несколько разновидностей моделей. Приведем пример наиболее известной модели, которую обычно используют сотрудники для описания своей компании (рис. 3.1). Второй способ

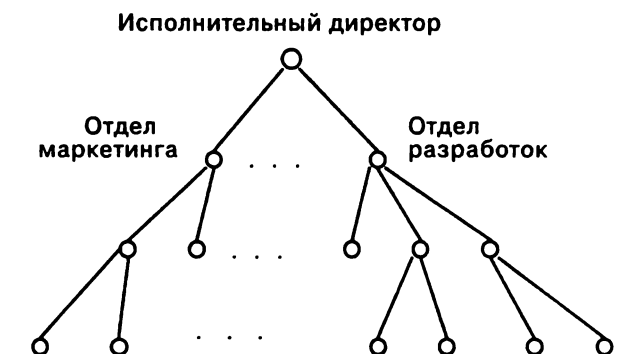


Рис.3.1. Модель иерархически организованной компании

описания компании состоит в указании ее различных функций, выполняемых отделами, совместная работа которых обеспечивает выполнение процесса (рис. 3.2). В этой модели заказчики, находящиеся вне компании, обслуживаются функциями внутри компании. Существует много других приемов и способов моделирования бизнеса.

Любая компания, как известно, - сложная система. Даже небольшую компанию нелегко увидеть во всех деталях. Более крупные компании фактически непостижимы: никто не имеет достаточно ясной картины, охватывающей абсолютно все детали. Поистине удивительно, что крупные компании могут довольно успешно функционировать. Впрочем, тому можно найти несколько причин. Самая главная из них состоит в том, что компании, несмотря ни на что, меняются довольно медленно (инерционно) и согласованно. По мере того как сами люди совершенствуются, происходят перемены в компании, причем лишь в той степени, в какой люди чувствуют себя способными осознать последствия. До тех пор пока компания не меняется радикально, свойственная ей инерция помогает ей функционировать. Но если компания встречается с трудностями и ей предстоят радикальные перемены, то существует серьезная опасность, что люди не

смогут как следует понять: “что они делают”, “почему они делают это”, “могут ли они делать что-либо другое”. Как правило, совет директоров поздно понимает, что дела плохи; но если ничего не делать, само существование компании может оказаться под угрозой. Обычно в такой ситуации компания расстается с исполнительным директором и руководство надеется, что его последователь сможет изменить дела к лучшему. Смена руководителя во многих случаях может изменить неправильный курс, но если проблемы, стоящие перед компанией, достаточно серьезны и глубоки, то изменений на уровне управленческого персонала будет недостаточно.

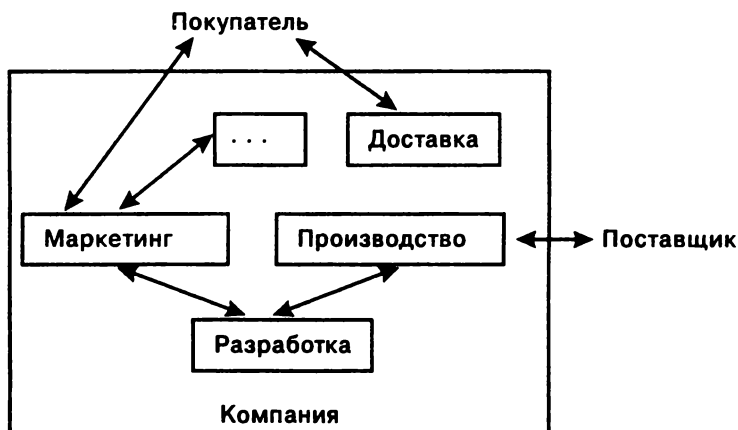


Рис.3.2. Модель, показывающая, как различные функции обеспечивают выполнение процесса

В качестве примера рассмотрим состояние дел в фирме IBM в 1993 г. Одна из крупнейших компаний мира была потрясена до основания, и никто не знал, чем это обернется для нее в будущем и как IBM будет выглядеть, когда все успокоится. Более десяти лет назад для большинства специалистов стала очевидной ведущая роль программного обеспечения, а не оборудования. В IBM не захотели признавать этот факт, продолжая цепляться за свою старую стратегию. Судя по всему, если бы фирма изменила курс лет десять назад, то о кризисе в 1993 г. не было бы и речи. А ведь в кризисной ситуации, как правило, не остается ничего другого, как надеяться на везение. Времени в обрез, и компания просто “истекает кровью”. Уже ничего нельзя делать систематически и приходится перестраиваться на ходу - проводить реинжиниринг на ощупь.

Так не должно быть. Работая более систематично с хорошими моделями бизнеса, можно гарантировать, что ваша компания постижима сверху донизу, по всей структуре. Можно описать компа-

нию, объяснить ее, прояснить новые мысли и идеи, сравнить их возможные последствия с тем, что есть в настоящий момент, разработать сценарии воздействия на компанию внешнего мира: заказчиков, конкурентов, продукции и т.д. IBM должна была моделировать различные сценарии развития и исследовать возможные результаты. Сценарии могли показать, что персональные компьютеры становятся все мощнее, потребность в мэйнфреймах существенно уменьшается, растет популярность систем “клиент-сервер” (которые на самом деле были известны еще в начале 1980-х гг.), требуются открытые решения и переносимое программное обеспечение и так далее. Оценка долгосрочных перспектив могла бы помочь фирме вовремя изменить курс и во всеоружии встретить наступающие перемены.

Итак, модель бизнеса нужна для того, чтобы управлять развитием компании систематически, а не полагаться на случай и везение. Очень важно, чтобы модель правильно акцентировала внимание на существенных факторах и скрывала несущественные. Но даже при наличии совершенной модели бизнеса фактор риска обязательно остается, равно как и неопределенность. Модели помогают у м е н ь ш и т ь р и с к, избежать некоторых ошибок и увеличить вероятность успеха. Ниже будут рассмотрены требования, предъявляемые к модели бизнеса с точки зрения различных категорий людей, взаимодействующих с компанией.

3.2

Требования к модели компании

Компания взаимодействует с различными категориями людей, и каждая категория (как мы упоминали в разд. 3.1) может иметь собственную модель компании. Рассмотрим только наиболее важные категории и кратко опишем их требования к модели компании. Отметим, что люди, относящиеся к одной категории, имеют единую точку зрения на модель, полностью определяемую информационными запросами этой категории.

3.2.1

Клиенты и партнеры

Фундаментальные изменения в компании не могут начаться до тех пор, пока каждое лицо из внешнего окружения компании не включится в процесс реинжиниринга (инновации). Заказчики и партнеры имеют относительно компании определенные ожидания, а компания, в свою очередь, ожидает нечто от них. Довольно часто

наиболее радикальные идеи по реинжинирингу приходят от клиентов или партнеров. Модель, включающая описание внешнего окружения, должна фокусировать внимание на том, как компания выглядит со стороны: что она может предложить своему окружению и наоборот. В этом контексте, несомненно, внутренняя организация работы компании не представляет какого-либо интереса для людей извне. С другой стороны, они жизненно заинтересованы в бизнес-процессах компании и их взаимодействиях с клиентами и партнерами. Интерес может вызвать и географический аспект: где расположена компания и какие процессы имеют место в том или ином регионе.

3.2.2

Исполнительный управленческий аппарат

Исполнительный управленческий аппарат формулирует перспективы и цели компании. Для этого он должен иметь ясную картину того, как эти факторы реализуются на практике, поэтому модели, которые использует управленческий аппарат, должны учитывать архитектуру компании. Эти модели дают менеджерам не только общую картину бизнес-процессов компании и их взаимодействия, но и подробно представляют каждый отдельный процесс. Для каждого бизнес-процесса менеджеры определяют цели, затраты, сроки, качество результата и т.д. Более того, они должны быть в состоянии распределять ресурсы (владельцев процессов, лидеров процессов, владельцев ресурсов), а также планировать бюджет и оценивать возможные последствия принимаемых финансовых решений.

3.2.3

Команда по реинжинирингу

Коллектив, проводящий реинжиниринг компании, должен иметь доступ к наиболее детализированным моделям. Членам команды по реинжинирингу нужны те же обобщенные модели, что и менеджерам, поскольку они должны при помощи этих моделей общаться с менеджерами. Однако им нужны и более подробные описания каждой стадии каждого из процессов. Ведь именно эти люди должны решать, какие виды ресурсов и в каких количествах потребуются, идентифицировать потенциальные узкие места и находить способы их устранения.

Здесь нужны не только полные описания, но и средства разработки соответствующих моделей (методологии и инструментальные средства - см. гл. 9). Команда должна быть в состоянии визуально представить образ будущей компании, понять и заново спроектировать ее на различных уровнях абстракции, построить прототип, определить все ресурсы, необходимые для ее реализации, и документировать ее. Методы и средства, используемые командой, должны обеспечивать (см. гл. 4):

① **Визуализацию образа будущей компании и окружающего ее мира.** Члены команды должны испытывать различные сценарии преобразования существующих бизнес-процессов, как в случае, когда компания берет на себя некоторые из задач, ранее решавшихся клиентами, так и в случае, когда клиенты привлекаются для решения задач, выполнявшихся до этого компанией. Команда должна работать с альтернативными архитектурами процессов и моделировать их воздействие на деятельность компании.

② **Описание альтернативных вариантов проекта после выбора архитектуры основного процесса компании.** В этом случае проектирование означает разработку бизнес-процессов так, чтобы они использовали человеческие ресурсы наиболее эффективным способом. Команда по реинжинирингу должна принимать решения относительно проектирования на различных уровнях, от общего архитектурного уровня (например, функциональная структура и типы ресурсов, которые должны быть в наличии) вплоть до детального, динамического уровня событий. Изучение бизнес-процессов в работе должно позволять выявлять конфликты, узкие места, тупики и несогласованности.

③ **Описание продукции компании в контексте того, как, когда и в ходе какого процесса она обрабатывается.** Каждый продукт и каждый тип продукта имеет жизненный цикл, который должен быть учтен в процессе моделирования.

④ **Адаптацию выбранного архитектурного решения к существующей компании.** Например, для компании, имеющей филиалы и отделения во многих странах, должна быть предоставлена возможность проводить реинжиниринг по филиалам поэтапно, не затрагивая, насколько это возможно, остальные части компании.

⑤ **Описание реализации конечного проекта с учетом как человеческих, так и технических ресурсов.**

⑥ **Представление реконструированной компании таким образом, чтобы каждый участник понял новую организацию работ, свои новые задачи и способы их выполнения,** т.е. модель должна быть понятна персоналу без длительного обучения и серьезного вмешательства в его работу.

3.2.4

Владелец процесса

Как правило, владелец процесса назначается из представителей исполнительного управленческого аппарата. Поэтому владелец процесса должен иметь ту же информацию, которая предоставляется исполнительному управленческому аппарату. Конечно, каждый владелец процесса обязан детально разбираться в своем процессе и принимать активное участие в его разработке. Кроме того, каждый владелец процесса должен иметь достаточно хорошее представление о смежных процессах.

3.2.5

Владелец ресурса

Каждый из ресурсов компании принадлежит некоторому владельцу, который должен иметь представление о бизнес-процессах и об их реализации с точки зрения человеческих ресурсов. Владелец ресурса должен уметь назначать каждому процессу адекватный ему вид ресурсов - специалистов с соответствующей профессиональной подготовкой, компетенцией, опытом и т.д.

3.2.6

Информационная система поддержки

Модель бизнеса - важнейшая исходная информация при разработке информационной системы поддержки (ИСП), поддерживающей бизнес-процессы компании. Как уже упоминалось выше, невозможно создать модель бизнеса компании, не разрабатывая одновременно (с точностью до некоторого сдвига во времени) поддерживающую информационную систему. Модели бизнеса и информационная система должны разрабатываться и н т е р а к т и в н о . Чтобы построить информационную систему, необходим доступ к большому объему информации о бизнесе. Например, необходимо знать:

- ① Всех пользователей (или ресурсы) как внутри компании, так и вне ее, которые будет использовать ИСП.
- ② Все бизнес-процессы и то, как каждый из них использует ИСП.
- ③ Типы документов, используемых в компании, и способы работы с ними. Короче, отдел информационного обеспечения обычно нуждается в такой же детальной информации, как и команда по реинжи-

нирингу. Специалисты в области информационных технологий в подавляющем большинстве уже привыкли использовать формальные методы в своей работе, поэтому они без возражений примут идею создания бизнес-модели компании.

3.3

Создание модели бизнеса

Наиболее известная модель бизнеса - иерархическая структура компании. Эта модель совершенно недостаточна для того, чтобы спроектировать и (или) изменить компанию. Вместо этого нужны модели, показывающие компанию в контексте ее клиентов, поставщиков, партнеров и т. д., т. е. модели, которые представляют бизнес-процессы компании и то, как она производит товары и услуги для внешнего мира.

Понять, как работает компания, - значит провести работу по обратному инжинирингу. Обычно это делается для того, чтобы получить прочную основу для кардинального улучшения различных аспектов компании в будущем. Модель существующей компании важна и тогда, когда требуется понять и объяснить, как функционирует компания или некоторый ее процесс. Например, если требуется при помощи реинжиниринга существенно улучшить временные характеристики, модель может выявлять узкие места, требуемые ресурсы и затраты на различных этапах процесса. Однако построение модели существующей компании дорого и требует много времени. В такой модели трудно добиться согласия. Следовательно, очень важно с самого начала определить, что необходимо моделировать и насколько подробно. Другими словами, **ключ к решению проблемы - это правильное сужение поля деятельности.** Следует сосредоточить внимание на тех частях бизнеса, которые должны быть радикально улучшены в результате реинжиниринга, т.е. важно проявить прагматизм и моделировать только то, что имеет значение для решения задачи реинжиниринга.

Описание новой компании - это работа по прямому инжинирингу, которая начинается с формулирования целей и образа (vision) будущей компании. После этого набрасываются различные сценарии. Для каждого сценария создается общее описание процесса, включающее заказчиков, поставщиков и т.д., а также сам процесс. Далее проводится имитационное моделирование различных процессов - при помощи деловой игры или компьютерной модели. Наконец, выбранная альтернатива реализуется.

Традиционные способы разработки моделей

Литература по реинжинирингу бизнес-процессов предлагает очень мало методик моделирования бизнеса. В [14] описана простая методика для представления последовательных процессов и упомянуто еще несколько методик. В [19] и [16] рассмотрена методика реинжиниринга, основанная на бизнес-событиях, но изложенные идеи не подкреплены конкретной технологией моделирования. Причина этого состоит в недооценке значения моделирования или (что более вероятно) в недоступности хорошей методики моделирования бизнеса. Действительно, все известные подходы к моделированию бизнеса принадлежат к одному семейству методов моделирования информационных сложных систем. Не вдаваясь в детали, приведем список наиболее известных подходов.

① Структурный анализ и структурное проектирование (Structured Analysis and Structured Design - SA/SD) [17, 20] является одной из самых известных методик разработки информационных систем. В методике SA/SD подчеркивается, что система предоставляет своим пользователям одну или несколько функций - так называемый *подход функциональной декомпозиции*. SA/SD предлагает набор средств, таких, как диаграммы потоков данных, диаграммы состояний-переходов, ER-диаграммы (на фазе анализа) и структурные схемы (на фазе проектирования).

② Методика IDEF (Integrated computer aided manufacturing DEFinition) была разработана BBC США на основе идей, появившихся в середине 70-х гг. На основе этой методики Министерство обороны США создало Федеральный стандарт обработки информации IDEF1X, который обеспечивает поддержку на нескольких уровнях посредством "модели бизнеса", "модели информационной системы" и "модели технологии". Моделирование бизнеса поддерживается ER-диаграммами для данных и диаграммами потоков данных специального вида, что позволяет иерархически описывать функции системы.

③ Методика SADT (Structured Analysis and Design Technique) [18] использует систему обозначений, похожую на диаграммы потоков данных в IDEF, для описания функций и структур данных информационной системы на основе декомпозиции.

Все эти методики, основанные на моделировании информационных систем, исходят из следующей парадигмы. При описании информационной системы предполагается, что она содержит два типа

сущностей: некоторый аналог программы (операционные сущности, которые выполняют некоторую обработку) и данные (пассивные сущности, которые хранят информацию, доступную для поиска, чтения и замены). Другими словами, информационная система описывается как некая абстракция компьютера.

При моделировании (разработке) сложные информационные системы разбиваются на составные части, каждая из которых рассматривается отдельно от других. Такой прием, как известно, называется *декомпозицией*. Классический подход к разработке сложных систем представляет собой структурное проектирование, при котором осуществляется алгоритмическая декомпозиция системы по методу “сверху-вниз”.

Жизненный цикл разработки сложной системы в этом случае складывается из этапов анализа, проектирования, программирования, тестирования и сопровождения, которые выполняются последовательно. Такой метод, называемый *каскадным*, имеет следующие отличительные особенности:

- ❶ *линейность выполнения этапов жизненного цикла разработки;*
- ❷ *четкое разделение данных и процессов их обработки;*
- ❸ *использование процедурных языков программирования.*

Недостатки каскадного метода сразу бросаются в глаза. Главный из них - последовательное выполнение этапов. Например, программирование можно начать только по завершении анализа и проектирования. Это приводит к большим потерям времени, не позволяет быстро разрабатывать прототипы программной системы. Каскадный принцип не согласуется с итеративным характером разработки программной системы, поскольку на последних этапах может выясниться необходимость внесения изменений в решения, принятые на предыдущих этапах [1, 11].

Для устранения этого недостатка Б.Бозм [10] предложил *спиральный подход*. Он заключается в том, что разработка проекта ведется как бы по спирали, причем на каждом ее витке выполняются последовательно перечисленные выше этапы, на которых уточняется проект [6]. Этот подход дополняет каскадный метод элементами итеративности. Но и для него характерен ряд существенных недостатков, к числу которых можно отнести [6]:

- трудоемкость внесения изменений;
- большой объем документации по проекту, затрудняющий программирование;
- серьезные ограничения возможностей сборки системы из готовых компонентов;
- сложность переноса на другие платформы.

Объектно-ориентированный подход к разработке моделей

Интерес к объектно-ориентированным технологиям значительно возрос в последние годы предыдущего десятилетия, когда в центре внимания разработчиков программного обеспечения оказались сложные информационные системы, не поддающиеся программированию “в лоб”. Создание подобных систем требует выполнения ряда этапов, предшествующих программированию.

Для того чтобы раскрыть сущность объектно-ориентированного подхода к разработке приложений, рассмотрим основные особенности сложных информационных систем [8].

3.5.1

Особенности сложных информационных систем

Иерархичность. Описывая характерные черты сложных систем, Г.Буч [1, 11] особое внимание уделяет их иерархическому характеру. Иерархическое построение таких систем облегчает понимание их человеком, возможности которого, связанные с восприятием информации, весьма ограничены. Иерархические структуры позволяют рассматривать только определенный уровень, не вдаваясь в детали реализации. Для сложной системы целесообразно моделировать два типа иерархии - типовую и структурную. Типовая иерархия отражает взаимосвязи “общее - частное”. В объектно-ориентированном подходе ей соответствует иерархия классов. Структурная иерархия показывает связи типа “это - часть того”. При объектно-ориентированном подходе ей соответствует иерархия объектов [8].

Групповая разработка. Разработка сложной информационной системы не может быть прерогативой одного человека. Для этой цели формируется группа, в которой каждый выполняет свои определенные функции. Иерархический характер сложных систем хорошо согласуется с принципом групповой разработки. В этом случае деятельность каждого участника проекта ограничивается соответствующим иерархическим уровнем.

Применяемые инструментальные средства (ИС) должны поддерживать групповую разработку. Для этого современные ИС реализуются на комплексах с архитектурой “клиент-сервер”. В них должна быть предусмотрена возможность интеграции результатов работы

отдельных участников проекта и защиты их от несанкционированного доступа.

Модифицируемость проекта. Сложные системы, имеющие достаточно долгое время жизни, обычно подвергаются многократной модификации. Это связано как с устранением ошибок, выявленных в процессе разработки, отладки или эксплуатации, так и с необходимостью внесения изменений и дополнений, вызванных изменениями внешних условий и требований к системе. Очевидно, что модификация сложных приложений может столкнуться с существенными трудностями ввиду значительного объема таких систем и большим числом взаимосвязей между их компонентами.

Сборочное проектирование. При разработке больших информационных систем широко используется концепция сборочного проектирования, основанная на идее повторно используемых компонентов [5]. Сборка прикладной системы из готовых компонентов позволяет значительно сократить время разработки. В связи с этим определяющее значение имеет то, насколько применяемые методики и поддерживающие их ИС обладают возможностями создания повторно используемых компонентов, а также насколько легко такие компоненты можно применять в других проектах.

Использование стандартных СУБД. Современные большие информационные системы обычно используют в работе стандартные СУБД в основном реляционного типа, причем реализация таких систем обычно осуществляется в среде “клиент-сервер”. Интеграция прикладной системы с базой данных (БД) ставит перед разработчиками ряд дополнительных задач. Главной из них является обеспечение преемственности, т.е. возможности использования в разрабатываемом приложении данных, накопленных в БД. Кроме того, при разработке приложения в большинстве случаев возникает необходимость проектирования логической структуры новой БД. Для интегрированных систем с клиент-серверной архитектурой используются специальные инструментальные средства.

Характерные особенности сложных информационных систем определяют требования, предъявляемые к методикам и инструментальным средствам, поддерживающим жизненный цикл их разработки (см. гл. 9).

3.5.2

Особенности объектно-ориентированного подхода

Стремление избавиться от недостатков структурного подхода привело к развитию новых идей, основанных на объектной декомпозиции. Такой подход к разработке программных систем получил название **объектно-ориентированного (ООП)**. В основе его лежат понятия *объект*, *класс*, *инкапсуляция*, *наследование* и *полиморфизм*

(см. ниже [1, 4, 11]). В реальном мире, а точнее, в интересующей разработчика предметной области, в качестве объектов могут рассматриваться конкретные предметы, а также абстрактные или реальные сущности. Например, о б ъ е к т а м и могут быть покупатель, фирма, производящая определенные товары, банк, заказ на поставку. Объект обладает *индивидуальностью* и *поведением*, имеет а т р и б у т ы , значения которых определяют его состояние. Так, конкретный покупатель, делая заказ, может оказаться в состоянии, когда денег на его счете не хватает для оплаты, а его “поведение” в этом случае заключается в “обращении в банк за кредитом”.

Каждый объект является представителем некоторого к л а с с а однотипных объектов. *Класс* определяет общие свойства для всех его объектов. К таким с в о й с т в а м относятся:

① состав и структура данных, описывающих атрибуты класса и соответствующих объектов;

② совокупность методов - процедур, определяющих взаимодействие объектов этого класса с внешней средой.

Например, описание класса “магазина” может включать некоторые атрибуты (индивидуальные для каждого объекта этого класса - конкретного магазина): “название”, “адрес”, “штат сотрудников”, “текущий счет”, а также методы: “формирование заказов на поставку товаров”; “передача товара со склада в торговую секцию” и т.д. Объекты и классы обладают характерными свойствами, которые активно используются при объектно-ориентированном подходе и во многом определяют его преимущества.

Инкапсуляция - скрытие информации [3]. При объектно-ориентированном программировании предусмотрена возможность запретить любой доступ к атрибутам объектов, кроме как через его методы. Внутренняя структура объекта в этом случае скрыта от пользователя, т. е. объекты можно считать самостоятельными сущностями, отделенными от внешнего мира. Для того чтобы объект произвел некоторое действие, ему извне необходимо послать сообщение, которое инициирует выполнение нужного метода. Инкапсуляция позволяет изменять реализацию любого класса объектов без опасения, что это вызовет нежелательные побочные эффекты в программной системе. Тем самым упрощается процесс исправления ошибок и модификации программ.

Наследование - возможность создавать из классов новые классы по принципу “от общего к частному”. Наследование позволяет новым классам при сохранении всех свойств классов-родителей (называемых в дальнейшем суперклассами) добавлять свои черты, отражающие их индивидуальность. С точки зрения программиста, новый класс должен содержать только коды и данные для новых или изменяющихся методов [3]. Сообщения, обработка которых не обеспечивается собственными методами класса, передаются суперклассу. Наследование позволяет создавать *иерархии классов* и является эф-

фективным средством внесения изменений и дополнений в программные системы.

Полиморфизм - способность объектов выбирать метод на основе типов данных, принимаемых в сообщении [2]. Каждый объект может реагировать по-своему на одно и то же сообщение. Полиморфизм позволяет упростить исходные тексты программ, обеспечивает их развитие за счет введения новых методов обработки.

Объектно-ориентированная декомпозиция заключается в представлении системы в виде совокупности классов и объектов предметной области. При этом иерархический характер сложной системы отражается в виде иерархии классов, а ее функционирование рассматривается как взаимодействие объектов.

При таком подходе сложная система описывается наиболее естественным образом.

Жизненный цикл объектно-ориентированной разработки программных систем содержит несколько этапов, но в отличие от структурного подхода в нем нет строгой последовательности их выполнения. Процесс носит принципиально итеративный характер, что полностью отвечает потребностям разработчиков (рис. 3.3).

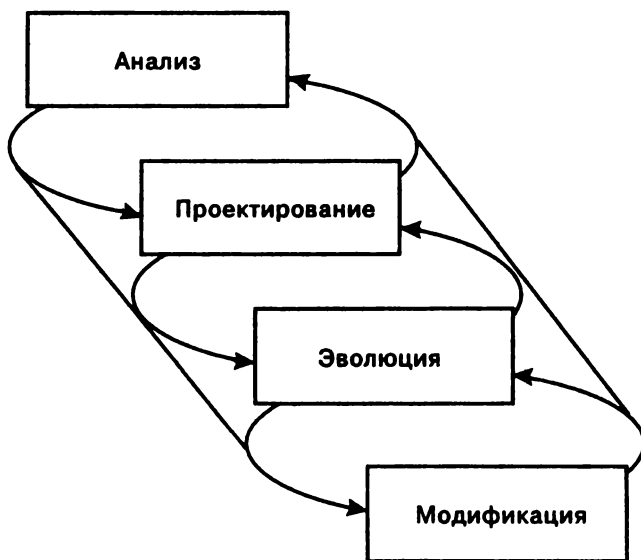


Рис. 3.3. Цикл разработки сложных систем с использованием объектно-ориентированного подхода

Цикл разработки приложения при использовании объектно-ориентированного подхода

Разработка начинается с этапа обследования - объектно-ориентированного анализа. Здесь определяются требования к системе. Затем осуществляется анализ предметной области, в ходе которого определяются основные классы и объекты, которые составляют словарь предметной области. Результатом обследования должны быть достаточно полные сведения для создания модели системы.

После обследования начинается объектно-ориентированное проектирование, в ходе которого детализируется представление классов и объектов, полученных на этапе анализа. Определяются структуры данных, методы, отношения между классами, разрабатываются сценарии взаимодействия объектов. При проектировании системы могут вводиться новые классы и объекты, если это потребуется для решения поставленных проблем. В результате проектирования должна быть создана детальная модель системы, составлены спецификации объектов, классов и отношений, достаточные для их программирования.

Программирование, тестирование и сборку системы Г.Буч [11] рассматривает как единый этап, называемый *эволюцией системы*. Объектно-ориентированный подход обеспечивает быстрое создание прототипов проектируемой системы, постепенное развитие которых приводит к конечному результату. На этом этапе также возможно введение новых классов, изменение структур данных, добавление новых методов. Следует заметить, что программирование и тестирование отдельных компонентов системы возможно до завершения проектирования, что экономит время разработки. Современные объектно-ориентированные ИС (см. разд. 9.3), применяемые при разработке программных систем, обычно обладают возможностями автоматизации ряда процессов, выполняемых на этом этапе. В частности, предусмотрена автоматическая генерация кодов программ.

Модификация системы может рассматриваться как отдельный этап. Возможность внесения изменений - естественное свойство сложных систем, обеспечивающее их развитие. При объектно-ориентированном подходе модификация не требует полного пересмотра проекта, затрагивая лишь необходимые для этого классы и объекты.

Главная особенность жизненного цикла при объектно-ориентированном подходе заключается в том, что нет строгой последовательности выполнения отдельных этапов. При разработке может выясниться необходимость дополнительного обследования; программирование и последующее тестирование могут потребо-

вать возврата к проектированию. Такой метод, названный Г. Бучем *возвратным*, отражает итеративный характер разработки приложения.

3.5.4

Преимущества и недостатки объектно-ориентированного подхода

Особенность процесса разработки современных сложных информационных систем состоит в том, что центр тяжести смещается от программирования к более ранним этапам - анализу и проектированию. Использование ИС, осуществляющих автоматическую генерацию кодов, возможность повторного применения проектных решений и сборки системы из готовых компонентов, значительно упрощает процесс программирования, поэтому эффективность принятых методик анализа и проектирования имеет решающее значение для судьбы проектов.

Преимущества объектно-ориентированного подхода

Распараллеливание работ. Как было отмечено выше, программирование и тестирование отдельных компонентов системы возможно до завершения проектирования, что экономит время разработки. При программировании может возникнуть необходимость внесения изменений в существующие классы или потребоваться введение новых объектов или классов. В этом случае, вернувшись к этапу проектирования или даже к анализу, можно внести изменения и дополнения, не подвергая проект полной переработке.

Упрощение внесения изменений. В отличие от структурного подхода, в объектно-ориентированном внесении изменений в проект имеет более локальный характер. В тех случаях, когда изменение носит характер уточнения, детализации, вводятся новые классы, наследующие поведение ранее созданных. *Наследование* - одно из основных свойств классов - позволяет в этих случаях не только не пересматривать ранее созданные объекты и классы, но даже обойтись без их повторной трансляции. В более сложных случаях, когда меняются методы, определяющие интерфейс классов, изменения в проекте будут более значительными, но и тогда они будут локализованы, затрагивая лишь классы, использующие эти методы.

Гибкая архитектура и переносимость. Объектно-ориентированная декомпозиция, в результате которой приложение представляется в виде совокупности классов и объектов, обеспечивает гибкость архитектуры системы. В клиент-серверной системе объекты могут

размещаться как на местах клиента, так и на сервере. В гетерогенных (разнородных) сетях возможна реализация классов на компьютерах разных типов, а фиксированный интерфейс каждого класса, определяемый набором его методов, обеспечит правильность функционирования системы. Изменения конфигурации оборудования не требуют внесения изменений в проект.

Повторное использование программных компонентов. Разрабатываемые в рамках некоторого приложения классы обычно отражают типовые решения, поэтому их использование возможно и в других приложениях. Возможность повторного использования программных компонентов - одна из самых сильных и привлекательных черт объектно-ориентированного подхода. Библиотеки классов, отражающие опыт в определенной области, позволяют значительно снизить объем программирования при разработке новых приложений. При наличии развитых библиотек классов проектирование и программирование новых приложений будет в основном сводиться к сборке системы из готовых компонентов.

Иерархический характер сложных программных систем позволяет значительно повысить эффективность повторного использования компонентов. “Чем более высокого уровня объекты вы можете повторно использовать, тем большего вы достигнете”, - отмечает Paul Fultove, директор информационных технологий US West Technologies [15].

Для того чтобы повторное использование компонентов принесло свои плоды, разработчики программных систем должны [7]:

- ① осознавать выгоды такого подхода;
- ② знать, какие части задачи могут быть решены с применением существующих программных средств;
- ③ заниматься поиском подходящих для повторного использования программ;
- ④ стремиться непременно найти такие программы;
- ⑤ использовать их даже в том случае, если они лишь частично совпадают с тем, что программист написал бы сам.

Следует заметить, что основные свойства классов и объектов - инкапсуляция, наследование и полиморфизм - полностью отвечают задаче повторного использования. В связи с этим в последние годы значительно возрос интерес к объектно-ориентированным ИС, которые предоставляют удобные услуги, связанные с повторным использованием программных компонентов [8].

Естественность описания. Объектно-ориентированный подход позволяет описывать как статические, так и динамические отношения между объектами модели. По описанию предметной области, выполненному на естественном языке, легко выделить объекты и статические связи между ними. Объекты соответствуют существи-

тельными, а связи - глаголам и отглагольным формам. Например, фраза "фирмы выполняют заказы" позволяет выделить классы объектов "фирма" и "заказ" и отношение между ними типа M:N, так как фирма может выполнять много заказов, а заказ может быть выполнен разными фирмами.

Кроме того, свойства наследования и инкапсуляции предоставляют возможность каждому участнику проекта рассматривать модель с удобным для него уровнем детализации. Руководители проекта могут работать с верхним уровнем модели, где отражаются только основные классы, объекты и связи. Другие разработчики или эксперты имеют возможность опускаться до более мелких, терминальных объектов, их свойств, связей, методов.

Недостатки объектно-ориентированного подхода

Объектно-ориентированная декомпозиция заметно отличается от алгоритмической, поэтому переход на новую технологию связан как с преодолением психологических трудностей, так и дополнительными финансовыми затратами. Здесь следует учесть и расходы на обучение методике, ИС и языку программирования. Для некоторых организаций эти обстоятельства могут стать серьезными препятствиями. Следует заметить, что это не является недостатком собственно объектно-ориентированного подхода, и в дальнейшем эти издержки будут компенсированы теми выгодами, которые сулят объектно-ориентированные технологии.

Недостатки самого объектно-ориентированного подхода лежат в области программирования. Рассмотрим основные из них. *Динамическое связывание*, предполагающее поиск метода в классе, которому принадлежит получающий сообщение объект, приводит к тому, что обращение к методу занимает в 1,75 - 2,5 раза больше времени, чем к обычной подпрограмме [1, 11]. Это, конечно, замедляет работу приложения. Однако, как указывает Г.Буч, динамическое связывание при использовании строго типизированных языков применяется примерно в 20% случаев от общего числа вызовов методов. В результате снижаются непроизводительные потери времени. В приложениях, где такие потери критичны, приходится прибегать к специальным программистским приемам.

Другой недостаток связан с излишней *многочисленностью методов и их вызовов*. Это объясняется тем, что для доступа ко многим атрибутам объектов (а к защищенным - всегда) используются специальные методы. Вызов метода высокого уровня абстракции приводит к тому, что в системе происходит каскад вызовов - от методов более высоких уровней иерархии к методам более низких уровней. Если время является ограничивающим фактором, такая ситуация может оказаться неприемлемой. Выход здесь видится в том, что по-

сле создания начального варианта системы производится ее оптимизация, связанная с уменьшением числа вызовов. Например, защищенные переменные можно сделать общедоступными и обращаться к ним напрямую, уменьшая тем самым число вызовов.

На компьютерах с сегментированной организацией памяти объектно-ориентированные системы при работе могут осуществлять *интенсивный межсегментный обмен*, что сказывается на их производительности. Это вызвано тем, что классы обычно объявляются в разных файлах и соответственно реализуются в разных сегментах. Проблема решается путем перераспределения классов по модулям; логическое описание модели не изменяется.

Для задач реального времени, выполняющихся в высоком темпе, нежелательным является *динамическое создание и удаление объектов*, что также активно используется в объектно-ориентированных языках. В [1] предлагается размещать такие объекты при создании программы, а не во время работы критичных по времени алгоритмов. Преодоление перечисленных затруднений связано с дополнительной работой программистов, но в то же время не требует очень больших усилий, так как действия, которые надо предпринять, достаточно очевидны. Кроме того, подобные проблемы возникают весьма редко. Следует также заметить, что объектно-ориентированные языки включают в себя средства, позволяющие достичь более высокого быстродействия программ по сравнению с традиционными языками [1, 11]. Таким образом, следует признать, что издержки объектно-ориентированного подхода с лихвой компенсируются его достоинствами.

Существует расхожее мнение, что объектно-ориентированный подход труден для понимания, поэтому переход на объектно-ориентированные технологии связан с большими затратами, которые не окупаются. В действительности дело обстоит по-другому. Традиционная и объектно-ориентированная технологии с точки зрения получаемых результатов по-разному ведут себя по отношению к затратам на их освоение. При использовании традиционных технологий некоторые результаты можно получить и при сравнительно небольших затратах, однако на определенной стадии наступает насыщение, когда даже значительные дополнительные затраты не приводят к существенному повышению эффективности.

Объектно-ориентированные технологии не дают немедленной отдачи. Эффект от их применения начинает сказываться после разработки двух-трех проектов и накопления повторно используемых компонентов, отражающих типовые проектные решения в данной области.

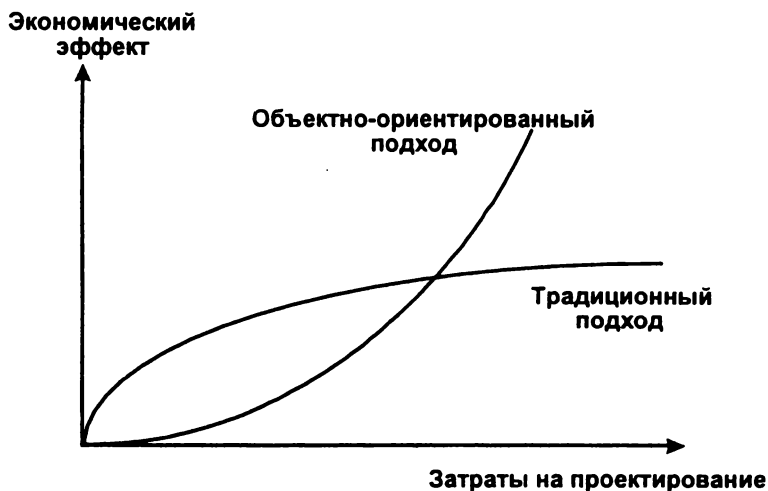


Рис. 3.4. Рост эффективности разработок по отношению к затратам при традиционном и объектно-ориентированном подходах

Как видно из рис. 3.4, при объектно-ориентированном подходе с приобретением опыта разработок кривая эффективности резко взмывает вверх за счет рассмотренных выше преимуществ, в особенности благодаря сборке систем из готовых программных компонентов.

Диаграмма на рис. 3.5 демонстрирует сокращение сроков разработок проектов. В обоих случаях есть стремление к определенным пороговым уровням. Но если для традиционного подхода снижение времени разработки связано в основном с ростом квалификации участников проектов, то при объектно-ориентированном подходе к этому прибавляется опыт использования типовых проектных решений.

К настоящему времени в мире имеется несколько десятков ИС, реализующих объектно-ориентированный подход к разработке сложных систем. Они используют разные методики анализа и проектирования. Практически все подобные средства обладают возможностями генерации кодов программ на объектно-ориентированных языках (C++, Ada, Smalltalk). Необходимо подчеркнуть, что ИС эффективны только в сочетании с методическим обеспечением.

Сроки разработки

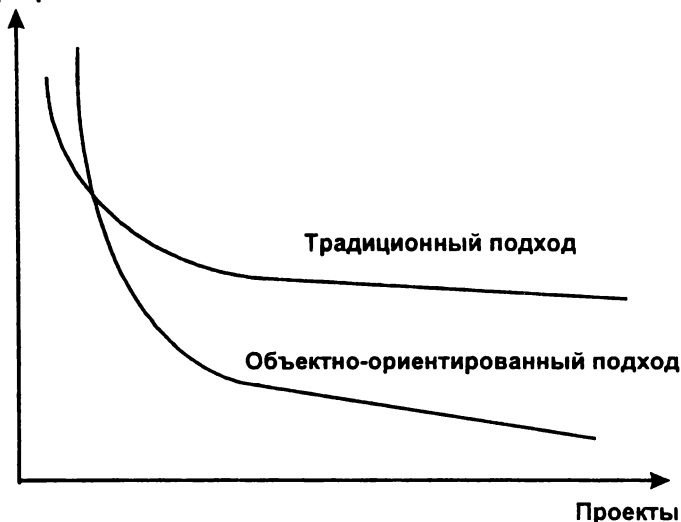


Рис.3.5. Снижение сроков разработки при традиционном и объектно-ориентированном подходах

3.6

Интегрированные подходы к разработке моделей

При всех достоинствах ОПП следует подчеркнуть, что он ориентирован на программистов, а не на менеджеров. Однако в проведении реинжиниринга участвуют специалисты двух типов - профессионалы в области реконструируемого бизнеса и разработчики информационных систем. Опыт реинжиниринга показывает, что по-настоящему успешное и новаторское внедрение информационных технологий является уникальным и творческим процессом: управляющие компаниями и специалисты-технологи, знакомясь с методами информационных технологий, сами делают открытия относительно возможностей их использования в своем конкретном бизнесе [12,13]. В то же время создание высококачественных информационных систем требует участия профессионалов в области ИТ. Возникает проблема нахождения общего языка. Решение этой проблемы стоит на пути интеграции таких современных технологий, как инженерия зна-

ний, объектно-ориентированное программирование, CASE-технологии, имитационное моделирование процессов и “активная” графика.

Именно такая тенденция и наблюдается в настоящее время в развитии методологий и инструментальных средств БПР (гл. 9).

Объектно-ориентированное моделирование признано базовой методологией БПР. Его особая роль объясняется следующим. Традиционно при создании информационных систем компании разработчики отталкивались от данных. В результате используемые ими подходы к моделированию систем были ориентированы на описание данных о сущностях реального мира и их взаимосвязей, но не на поведение этих сущностей. Поскольку реинжиниринг ориентирован на процессы, а не на данные, традиционные подходы оказались неадекватны. В настоящее время только объектно-ориентированный подход позволяет описывать как данные о сущностях, так и их поведение. Кроме того, он обеспечивает создание прозрачных, легко модифицируемых моделей бизнеса и информационных систем, допускающих повторное использование отдельных компонентов.

Однако модели, создаваемые в соответствии с этой методологией, довольно сложны, и мало вероятно, что управляющие компаниями могут работать с ними так же естественно и легко, как профессионалы в области ИТ.

CASE-технологии использовались в реинжиниринге практически с самого начала. Однако их ориентация на разработчиков информационных систем привела к тому, что в настоящее время их начинают объединять с другими современными технологиями, в первую очередь с объектно-ориентированными.

Имитационное моделирование обеспечивает наиболее глубокое представление моделей для непрограммирующего пользователя, а также наиболее полные средства анализа таких моделей. Модели создаются в виде потоковых диаграмм, в которых представлены основные рабочие процедуры в компании и описано их поведение, а также информационные и материальные потоки между ними. Однако построение реальных имитационных моделей - довольно трудоемкий процесс, а их детальный анализ (выходящий за рамки простого сбора статистики по срокам и стоимостям) зачастую требует от пользователя специальной подготовки. Для описания рабочих процедур может понадобиться дополнительное программирование. Таким образом, при попытке привлечь менеджеров к непосредственному использованию средств имитационного моделирования возникают определенные проблемы.

Эти проблемы преодолеваются с помощью методов *инженерии знаний* (гл. 9). Благодаря им можно непосредственно представлять в моделях плохо формализуемые знания менеджеров о бизнес-процессах, в частности рабочих процедурах. Кроме того, решается проблема быстрой разработки приложений и создания интеллекту-

проблема быстрой разработки приложений и создания интеллектуального интерфейса конечного пользователя со сложными средствами анализа моделей.

Средства быстрой разработки приложений позволяют сокращать время создания поддерживающих информационных систем и, следовательно, необходимы не только в ходе реинжиниринга компании, но и на этапе эволюционного развития, сопровождающегося постоянными модификациями и улучшениями информационных систем компании.

В настоящее время начинают появляться интегрированные многофункциональные средства, объединяющие в себе различные современные технологии (разд.9.3 и Приложение 1).

ЛИТЕРАТУРА

1. Б у ч Г. **Объектно-ориентированное проектирование с примерами применения** /Пер с англ. - М.: Конкорд, 1992.
2. Д у н к а н Р. **Замещение операторов и функций в Си и Си++** //PC Magazine /USSR/. - 1991. - №3.
3. Д у н к а н Р. **Инкапсуляция данных и наследование свойств в Си++** //PC Magazine /USSR/. - 1991. - №3.
4. Д у н к а н Р. **Си++ - новое мышление в программировании** //PC Magazine /USSR/. - 1991. - №3.
5. Л и п а е в В. В., П о з и н Б. А., Ш т р и к А. А. **Технология сборочного программирования** /Под ред. В.В.Липаева. - М.: Радио и связь, 1992.
6. Метод "по спирали" быстро ведет к цели //Деловой мир. - 1995. - № 23-24.
7. Н о в о ж е н о в Ю. В. **Объектно-ориентированные технологии разработки сложных программных систем.** - М.: Издание А.С., 1996.
8. Т а л б е р т Дж., Б е р н с т е й Б. **Программы многократного использования становятся реальностью. Making reuse a reality**//Компьютерра. - 1995. - №18.
9. Ш а п о т М. Д. **Инструментальные средства поддержки реинжиниринга бизнес-процессов**//Материалы семинара "Динамические интеллектуальные системы в управлении и моделировании". - М.: ЦРДЗ, 1996.
10. В о e h m В. **A spiral model of software development and enhancement** //IEEE Computer. - 1988. - № 25(5).
11. B o o c h G. **Object-Oriented Analysis and Design with Applications** //Bengamin/Cummings, Redword City, CA, USA, 1994.
12. H a m m e r M. and C h a m p y J. **Reengineering the Corporation: A Manifesto for Business Revolution.** - N-Y: HarperCollins, 1993.
13. J a c o b s o n I., E r i c s s o n M., J a c o b s o n A. **The Object Advantage: Business Process Reengineering with Object Technology**//ACM Press. - N-Y: Addison-Wesley Publishing, 1995.
14. J o h a n s s o n H., M c H u g h P., P e n d l e b u r y J. and W e e l e r W. **Business Process Reengineering. Breeakpoint Strategies for Market Dominance.** - Chichester: John Wiley & Sons, 1993.

15. M o a d J . **Object Methods Tame Reengineering Madness**//Datamation. - May, 1995.
16. R o s s D . **Applications and extensions of SADT**//IEEE Computer. - April, 1985.
17. S h e l e g W . **Business Process Reengineering driven by business events**// Database Newsletter. - September/October, 1993. -21(5).
18. T s a n g E . **Business Process Reengineering and why it requires business event analysis**//Case Trends. - March, 1993.
19. Y o u r d o n E . and C o n s t a n t i n e L . L . , **Structured Design: Fundamentals of a Discipline of Computer Program and System Design**. - Englewood Cliffs, N-Y: Prentice-Hall. - 1979.
20. Y o u r d o n E . **Modern Structured Analysis**. - Yourdon Press/Prentice-Hall. - 1989.

4

ОСНОВНЫЕ ЭТАПЫ РЕИНЖИНИРИНГА БИЗНЕС-ПРОЦЕССОВ

4.1

Структура и этапы реинжиниринга

В каждой компании можно выделить “ресурс”, который отвечает за разработку и сопровождение бизнес-процессов компании. В небольших компаниях этот ресурс может явно не выделяться, а входить в руководящий аппарат. Обычно этот ресурс называют группой разработки бизнеса (бизнес-процессов). Эта группа на входе имеет новые цели, на выходе - измененную компанию (рис 4.1).

Разработка бизнеса - это не обычный бизнес-процесс. Его особенность в том, что у него нет внешнего потребителя. Его можно рассматривать как внутренний процесс, но многие специалисты по

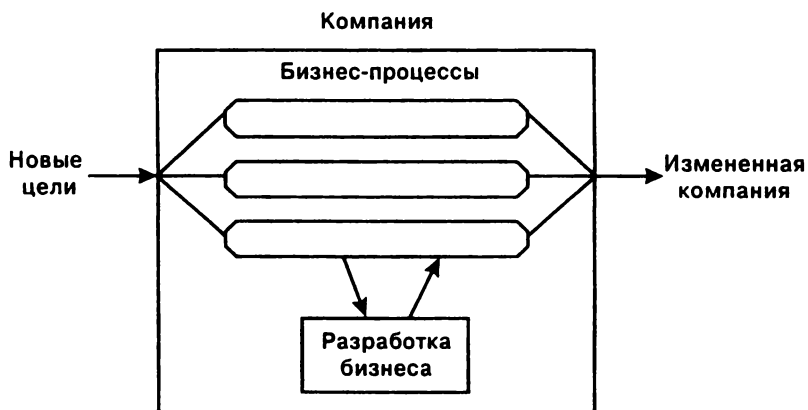


Рис. 4.1. Группа разработки бизнеса

БПР стараются избегать использования понятия “процесс” при описании внутренних организационных шагов. Будем рассматривать разработку бизнеса как объект, точнее, как агрегат объектов. Разработка бизнеса может быть частью различных задач компании. Среди этих задач - реинжиниринг бизнеса (одна из наиболее интересных задач) и усовершенствование бизнеса.

Проект по реинжинирингу бизнеса обычно включает следующие четыре этапа (рис. 4.2):

① **Разработка образа будущей компании** - спецификация основных целей компании исходя из ее стратегии, потребностей клиентов, общего уровня бизнеса в отрасли (определяется на основе анализа какой-либо из ведущих фирм смежной отрасли, не являющихся конкурентами и готовых предоставить необходимую информацию о себе) и текущего состояния компании.

② **Создание модели существующей компании (называемое также обратным, или ретроспективным инжинирингом).** На этом этапе менеджеры с участием разработчиков информационных систем должны разработать детальное описание существующей компании, идентифицировать и документировать ее основные бизнес-процессы, оценить их эффективность.

③ **Разработка нового бизнеса (прямой инжиниринг).**

3.1. Перепроектирование бизнес-процессов. Создание более эффективных рабочих процедур (элементарных заданий, из которых строятся бизнес-процессы), определение способов использования информационных технологий, идентификация необходимых изменений в работе персонала.

3.2. Разработка бизнес-процессов компании на уровне трудовых ресурсов. Здесь проектируются различные виды работ, подготавливается система мотивации, организуются команды по выполнению работ и группы поддержки качества, создаются программы подготовки специалистов и т.д.

3.3. Разработка поддерживающих информационных систем. На этом этапе определяются имеющиеся ресурсы (оборудование, программное обеспечение) и реализуется специализированная информационная система (или системы) компании.

④ **Внедрение перепроектированных процессов.** Интеграция и тестирование разработанных процессов и поддерживающей информационной системы, обучение сотрудников, установка информационной системы, переход к новой работе компании.

Необходимо подчеркнуть, что перечисленные этапы выполняют не последовательно, а частично параллельно, причем некоторые этапы повторяются. Для разработки общего представления будущего бизнеса надо сначала разобраться в деятельности существующей

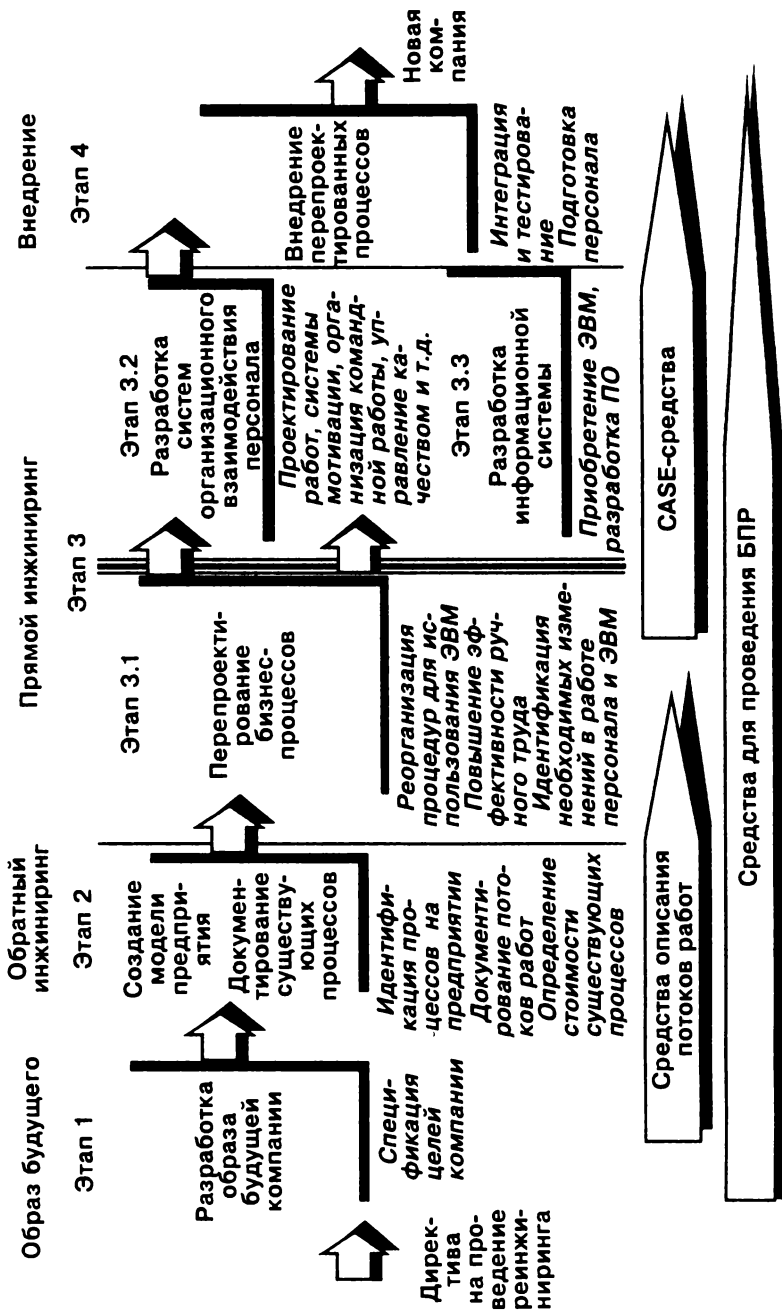


Рис.4.2. Общая схема реинжиниринга

компании. Таким образом, если считать перечисленную выше последовательность этапов моделью, то эта модель весьма приближительна. Она показывает работу по реинжинирингу бизнеса, выполняемую в контексте разработки бизнеса, и подчеркивает тот факт, что реинжиниринг состоит из двух основных шагов: обратного и прямого инжиниринга новой компании. Интуитивно (и несколько упрощенно) можно утверждать, что:

**РЕИНЖИНИРИНГ БИЗНЕСА = ОБРАТНЫЙ ИНЖИНИРИНГ БИЗНЕСА +
ПРЯМОЙ ИНЖИНИРИНГ БИЗНЕСА**

Анализ существующей компании - это работа по обратному инжинирингу. Предполагается, что здесь построится абстрактная модель бизнеса и процессов, которые необходимо улучшить. Разработка новой компании - это прямой инжиниринг нового бизнеса.

Проект по реинжинирингу может быть инициирован различными способами и в различных ситуациях. Как правило, проект начинается в ответ на директиву, которая объясняет, почему что-то надо менять, и определяет цели, которые надо достичь. Эта директива запускает процесс, называемый “образ будущего” (визуализация), который строит общую схему бизнеса новой компании. Конечно, для этого надо знать стратегию компании и понимать настоящее положение дел. Как минимум, необходимо понимать компоненты, которые пересматриваются, чтобы сконцентрироваться на сущностях, которые будут изменяться. Это очень важно для того, чтобы внести радикальные перемены в бизнес без излишнего риска. Результатом данного этапа является то, что называют “спецификацией целей компании”, то есть взгляд на будущий бизнес.

Таким образом, этап визуализации запускает работу по инжинирингу существующего бизнеса, в результате чего создается модель существующего бизнеса. В идеальной компании, где люди понимают, как важно иметь ясные и хорошо документированные процессы, модель существующего бизнеса уже имеется и служит основой для работы по прямому инжинирингу. Эта модель была создана, когда компания последний раз подвергалась реинжинирингу, и нет необходимости делать эту работу снова. Однако будем исходить из того, что большинство компаний еще не продвинулось так далеко, и они должны начать с моделирования существующей компании.

Работа по инжинирингу нового бизнеса предполагает создание одного или более новых процессов, их проектирование, разработку поддерживающей информационной системы и т.п. с целью получения модели перепроектированной компании. При этом в качестве исходных данных используется спецификация целей компании. Результат всего проекта по реинжинирингу в решающей степени зависит от этого этапа.

Проект по реинжинирингу завершается этапом внедрения перепроектированного бизнеса. На этом этапе перепроектированная компания должна быстро и органично внедриться в реальную организацию.

Обращаем внимание на то, что перечисленные этапы работ по реинжинирингу представляют собой лишь один из вариантов для иллюстрации основной идеи. Реинжиниринг можно выполнять различными способами, и следует выбирать методы, которые лучше подходят для вашей компании, принимая во внимание ее размер, природу ее бизнеса, степень знакомства персонала с моделированием и т.д.

Еще раз подчеркнем, что выполнение работ по инжинирингу является итеративным процессом. На различных этапах могут быть обнаружены способы, позволяющие функционировать более эффективно, чем это представлялось ранее. Когда такое случается, необходимо вернуться назад и изменить спецификации целей компании. Открытия такого плана могут быть обнаружены весьма поздно, даже после того, как проект уже оттестирован. В этом случае цена итерации гораздо больше, чем на ранних этапах работы. Итерация внутри нескольких этапов, так же как и на одном этапе, не является чем-то необычным. Однако, если сначала думать, а потом делать, можно уменьшить риск задержек и дорогостоящих переделок. Хорошие методы моделирования и инструментальные средства значительно уменьшают риск. Но если все же приходится возвращаться, важно быть последовательным в обновлении моделей, чтобы отразить перемены, привнесенные итерацией.

Когда компания проводит реинжиниринг и внедряет новую модель бизнеса, она вступает в новую фазу своей жизни. Предположим, что реинжиниринг прошел успешно, и теперь компания имеет хорошо документированные процессы (по крайней мере, новые процессы). Что теперь? Остановиться на этом? Расслабиться? Ни в коем случае! Например, компания SAS, достигнув первых успехов, не была готова ко второму этапу. «Недостаток новых целей быстро оказал отрицательное влияние на SAS. Отлично налаженная гармония начала расползаться по швам. Вновь проявились все деструктивные силы. Вместо одной общей цели были поставлены отдельные цели разных групп и личностей» [1]. Изменение компании - это непрерывно продолжающийся процесс. После первой попытки реинжиниринга (которая, возможно, затронула только часть бизнеса) наступает следующая фаза, за ней следующая и так далее. Похоже, что первый проект по реинжинирингу требует больше внимания руководства, однако зрелая компания всегда будет уделять внимание реинжинирингу.

В действительности, как правило, между первым и последующими проектами по реинжинирингу разница невелика. Основные различия относятся к работам по усовершенствованию бизнеса. Большие, радикальные перемены называются реинжинирингом, в то время как непрерывные изменения меньшего масштаба называют усовершенствованием бизнеса. Реинжиниринг - это не разовое усилие, он делается много раз за время жизни компании. Он проводится для процессов, не включенных в первый проект и для уже реконструированных процессов в соответствии с новыми запросами клиентов, новым уровнем конкуренции, нововведениями в процессе или методике. Группа по разработке бизнеса - это команда, которая занимается как усовершенствованием бизнеса, так и реинжинирингом. Все виды деятельности, описанные в этой главе, - разработка образа будущего бизнеса, инжиниринг существующего бизнеса, конструирование нового бизнеса, внедрение нового бизнеса - являются частью разработки бизнеса.

Как правило, компании не могут проводить обновление слишком часто. Работники могут не справиться с непрерывным потоком изменений. Кроме того, требуется время для хорошей проверки нововведений перед их внедрением в большем масштабе. Лучше некоторое время собирать предложения и представлять их, скажем, раз в год, если они не слишком срочные. Эффект внедрения чего-то, что еще не готово к внедрению, может быть весьма разрушительным. Помимо серьезного дисбаланса бизнеса, незрелое нововведение может подорвать доверие персонала и участников к людям, проводящим изменения. Это снижает эффективность их работы в будущем. Кроме того, требуется время для внедрения улучшений, требующих изменения системы компьютерного сопровождения компании. Этой постоянной работой по усовершенствованию следует руководить как единым целым. Собирайте предложения по усовершенствованию, сначала не думая о том, как и когда они будут реализованы. Предложения следует скомбинировать в функционально связанные и полностью интегрированные "версии", независимо, будут они реализованы техническими или людскими ресурсами. Важное средство в управлении работой по усовершенствованию - *метрики*. Работа по усовершенствованию должна основываться на том, насколько достигнуты цели. Следовательно, для каждого процесса необходимо установить измеримые цели. Если измеренные значения показывают, что процесс или подпроцесс не достигает своей цели, с ним, вероятно, что-то не в порядке.

Основания для начала работ по реинжинирингу

Директива обычно инициирует усилия по реинжинирингу. Директива должна быть составлена в терминах высокого уровня. Ее общий тон должен выражать ожидания от реализации проекта. Чтобы стимулировать существенные изменения в бизнесе, она должна уделить внимание серьезным проблемам и показать, какие радикальные перемены ожидаются в будущем. Директива должна объяснить ситуацию, в которой находится компания, и почему в этом положении нельзя оставаться. Говоря конкретно, директива должна представить документ типа “аргументы для действий” (case-for-action paper), как называет его М.Хаммер [3]. Такой документ объясняет, почему бизнес должен быть реконструирован. Директива должна явно и недвусмысленно объяснить следующие моменты:

- Окружение компании. Кто является клиентами и конкурентами компании? Как окружение изменяется в настоящий момент?
- Ожидания клиентов. Почему компания должна удовлетворять нужды клиентов иначе, чем сейчас?
- Увеличение конкуренции. Как конкуренты удовлетворяют нужды клиентов?
- Трудности бизнеса компании. Что делается неправильно? Почему это делается так?
- Диагноз компании. Почему компания должна фундаментально переосмыслить методы работы и радикально изменить ее операции?
- Риск, вызванный сохранением существующего положения. Каковы последствия непроведения реинжиниринга?

Директива должна быть откровенной, однако она не должна преувеличивать серьезность ситуации. Как правило, проблемы уже широко известны и распознаны, хотя, возможно, и не выражены ясно и кратко. Абсолютно необходимо, чтобы руководство компании было полностью согласно с директивой и поддерживало ее. Работники должны быть убеждены, что компания не может продолжать идти прежним курсом, если она хочет выжить и остаться конкурентоспособной. Они должны понять, что с проблемами можно справиться, но надо уже сейчас предпринимать для этого значительные усилия.

4.3

Разработка образа будущей компании

Цель этого этапа - выработать взгляд на новую компанию и сформулировать его в терминах спецификации целей компании. Работа по визуализации новой компании не формализована так, как работа по созданию модели существующей или новой компании. Т.Давенпорт [2] дает довольно подробное описание этого этапа, на котором следует (рис. 4.3):

- обеспечить соответствие стратегических целей компании целям новой компании;
- интервьюируя клиентов, выявить, как можно более полно удовлетворить требования клиентов в продуктах, услугах и процессах компании;

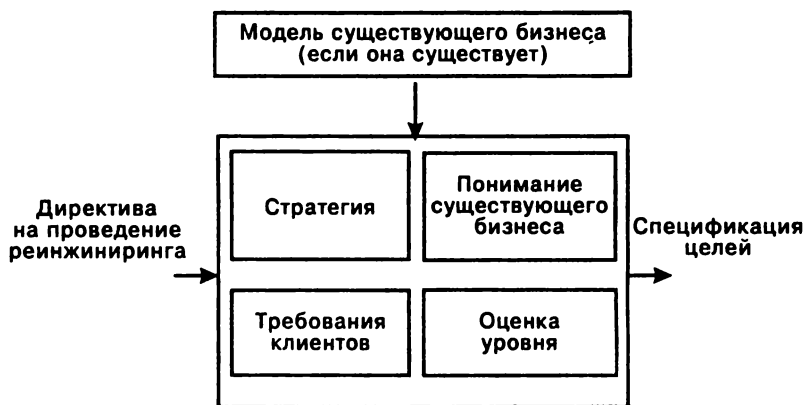


Рис. 4.3. Разработка образа будущей компании

- понять, как функционирует существующая компания, с тем, чтобы специфицировать новую компанию;
- сравнить компанию с другими предприятиями из ее окружения при помощи оценки уровня.

В команду по разработке образа будущего должны быть включены люди различных специальностей. Наиболее важно присутствие в команде представителей основных клиентов компании. Другие участники - это люди из самой компании и, возможно, консультант по реинжинирингу. Удобно построить работу как серию рабочих совещаний с последующей работой отдельных специалистов или маленьких групп (см. подразд. 4.8.3).

4.3.1

Стратегия

Стратегия бизнеса компании должна соответствовать способу ее работы, то есть должна существовать строгая связь между стратегией и процессами, которые использует компания. Стратегия должна соответствовать долгосрочным целям компании и ориентировать процессы на эти цели. Как писал Т.Давенпорт [2], *“Стратегия и цели процессов должны усиливать друг друга и звучать в унисон”*. Т.Давенпорт [2] сформулировал требования, которым должна удовлетворять стратегия новой компании:

- Стратегия не должна основываться исключительно на финансовых целях. Как правило, сотрудники компаний считают финансовые цели недостаточно конкретными, поскольку они не видят, как они могут достичь этих целей.
- Стратегия должна быть сформулирована так, чтобы ее эффект можно было измерить.
- Стратегия должна фокусироваться на ограниченной и реальной идее бизнеса.
- Стратегия должна вдохновлять, а не принуждать работников каждого уровня создавать компанию, соответствующую поставленным целям.

4.3.2

Требования клиентов

Очевидно, что наибольший импульс на улучшение бизнеса исходит от клиентов компании. Следовательно, необходимо анализировать и количественно оценивать ожидания (настоящие и будущие) клиентов компании. Работники компании очень часто недооценивают важность такого анализа. Они считают, что знают все ответы, и бывают очень удивлены, когда оказывается, что это не так. Анализ лучше всего проводить при помощи опроса клиентов, в ходе спонтанных интервью или систематических исследований. Цель заключается в том, чтобы удовлетворить потребности клиентов по расширению и улучшению продуктов и услуг. Интервью обнаруживают процессы, нуждающиеся в наиболее срочном улучшении. Х.Юхансон [6] рекомендует применять для выяснения нужд и желаний потребителей методы маркетинговых исследований.

4.3.3

Понимание существующего бизнеса

Для разработки образа новой компании необходимо понять, как выглядит существующая компания. В компании, где применялся бизнес-инжиниринг, эта модель уже существует. Если нет, то ее следует создать на этапе обратного инжиниринга. На этом этапе необходимо рассмотреть только те части бизнеса, которые подлежат реинжинирингу. Таким образом, ясно, что работа по визуализации новой компании начинается до и кончается после работы по обратному инжинирингу существующего бизнеса. Следовательно, визуализация и обратный инжиниринг существующего бизнеса - это параллельные работы.

4.3.4

Оценка уровня (benchmarking)

Оценка уровня - это методика, которая все чаще используется вместе с реинжинирингом, особенно как средство визуализации того, как будет функционировать новая компания. При оценке уровня осуществляется сравнение новой компании с лучшими фирмами и анализ информации с целью убедиться, что поставленные цели превосходят аналогичные у конкурентов. Следует сосредоточить внимание на компаниях, которые работают в вашей отрасли, и на компаниях в других областях, использующих подобные процессы. Выберите компании, которые отвечают следующим требованиям:

- имеют хорошую репутацию;
- полностью удовлетворяют нужды пользователей;
- производят товары хорошего качества;
- являются признанными лидерами в своей области;
- интересны с точки зрения оценки уровня.

4.3.5

Выполнение этапа визуализации

Стратегия является основой, с помощью которой команда по реинжинирингу формирует образ будущей компании. Стратегия должна согласовываться с целями приоритетных процессов и с целями, вытекающими из стратегии. Стратегия должна быть четко сформу-

лирована в строгих терминах, без банальностей. Знания о том, как функционирует компания и ее процессы, очень важны при создании новых процессов.

Команда по реинжинирингу начинает работу по визуализации с того, что решает, какой процесс или процессы наиболее важны. Конечно, очень важно сконцентрироваться на тех местах, где возможны скорейшие позитивные результаты. Не следует слишком распылять силы, целесообразно сосредоточиться на повышении эффективности жизненно важных процессов. М.Хаммер [3] дает более детальное описание того, как выбирать процессы, наиболее подходящие для реинжиниринга. Он определяет три критерия:

❶ Выбирайте процессы, которые наиболее беспокоят компанию. М.Хаммер приводит интересный список симптомов, характеризующих такие процессы.

❷ Выбирайте процессы, которые являются жизненно важными для клиентов и центральными для существования всей компании.

❸ Выбирайте процессы, реинжиниринг которых имеет хорошие шансы на успех.

При помощи оценки уровня постарайтесь узнать, как работают другие компании, изучайте работу своих “лучших конкурентов” и “лучших коллег”. Имея эту информацию, можно поставить достойные цели и понять, как их достичь. Визуализация нового бизнеса - это итеративная задача. Следует потратить столько времени, сколько нужно на тестирование альтернативных сценариев процессов, проверку каждого процесса, на внимательное осмысление того, что уже сделано, и так далее. Для проверки на деле предлагаемых идейстройте прототипы. Именно при прототипировании созидательная способность группы расцветает и приносит плоды. Никакое предположение не должно считаться слишком эксцентричным до его опробования; в конце концов, именно странные идеи дали начало фундаментально новаторским решениям.

Концентрируйте внимание на наиболее важных процессах. Рассматривая их в порядке важности, делайте наброски и погружайтесь в интерфейс между ними и их окружением. Является ли данный интерфейс фиксированным или его можно модифицировать? Может ли область действия процесса быть расширена так, чтобы работа, выполняемая раньше клиентом (или не выполняемая никем), выполнялась теперь компанией? Может ли область действия процесса быть уменьшена так, чтобы работа, выполняемая раньше компанией, выполнялась теперь клиентом? Процесс считается улучшенным, если он лучше обслуживает заказчика, стал проще, дает лучшую продукцию, работает быстрее и так далее. Для каждого процесса установите измеримые цели, относительно которых можно измерять реконструированную компанию. Позднее, когда новая компания будет создана, эти цели можно будет использовать, чтобы измерять, как улучшилось функционирование процессов.

Команда должна сделать грубую оценку, какие виды поддержки (особенно в области ИТ) потребуются для новых процессов. Важно выявить на ранней стадии, какие методики доступны для реализации бизнеса. Нужно ли для поддержки процесса использовать новое заказное программное обеспечение? Следует ли применять существующие программы или имеет смысл приобрести готовые? Сможете ли вы изыскать ресурсы, внутренние или внешние, для создания программного обеспечения? Существенна ли имеющаяся конфигурация компьютерных систем, терминалов, рабочих станций, сетей? Требуется ли совместимость с существующей информационной системой (см. гл. 8)?

Результат работы по визуализации - это спецификация целей компании, которая представляет образ новой компании.

4.4

Спецификация целей компании

Спецификация целей представляет описание образа будущей компании. Она должна:

- Идентифицировать и назвать новые или радикально измененные бизнес-процессы реконструированной компании.
- Содержать общее высокоуровневое описание будущих процессов, подчеркивая, чем они отличаются от настоящих. Для каждого процесса спецификация должна определить клиента, поставщика или других партнеров. Она должна описывать входы, действия и продукцию каждого процесса. Эти описания не должны быть детальными, они предназначены для того, чтобы стимулировать дискуссию о будущей компании среди руководства и клиентов. Описания должны представлять в откровенных выражениях философию компании, ее намерения относительно новых и измененных процессов, и как ими предполагается управлять. Чтобы прояснить сложные моменты в компании, эти описания должны быть хорошо написаны, они могут включать простые и понятные заметки об отношениях процессов с их окружением. Следует избегать разъяснения очевидных вещей, что обычно означает неполноту описаний. Детали и глубина оценок могут быть отложены до этапа, который называют прямым инжинирингом нового бизнеса.
- Определять измеримые свойства и цели для каждого процесса: цена, качество, жизненный цикл, время и удовлетворение пользователя. Все цели должны быть соотнесены со стратегией. Цели должны быть количественными, чтобы мотивировать команду по реинжинирингу работать лучше, чем требуется для совершения простых количественных изменений.

- Определять технологии, которые будут поддерживать процессы, уделяя особое внимание ИТ. В этом контексте спецификация должна также установить, будет ли использоваться существующая информационная система компании, и если будет, то задать требования к совместимости.

- Описывать воображаемые сценарии будущего. Спецификация должна, насколько возможно, предсказывать, как процессы будут изменяться в ближайшие несколько лет по причине новых технологий, новых интерфейсов с окружением и других типов ресурсов. Также следует выяснить, будут ли процессы нуждаться в адаптации к другим организациям, таким, как филиалы или партнеры того или иного рода. Одинаковые (или похожие) процессы, возможно, могут применяться в различных географических регионах. Однако детализацию не следует доводить до абсурда. С одной стороны, не следует строить жесткую структуру процесса, если можно предположить, что его придется менять. С другой стороны, не следует вносить в процесс ненужную гибкость, т.е. гибкость, которая не будет использована.

- Содержать список критических факторов успеха (см. разд. 1.3).

- Описывать факторы риска, которые должны быть устранены. Определять возможные ловушки, угрожающие работе по реинжинирингу, и содержать рекомендации, как их избежать (см. разд. 1.3).

4.5

Разработка модели существующего бизнеса

Здесь данный этап только упоминается, чтобы обозначить его место в общем контексте; подробно он будет обсуждаться в гл. 6. Работа по инжинирингу существующего бизнеса начинается, как только определен образ будущей компании и выявлены приоритетные процессы. На этом этапе хотелось бы получить ясную картину того, как процессы работают сейчас, до проведения реинжиниринга. Необходимо убедиться в том, что известно, что имеется и что хотят изменить. Обратный инжиниринг помогает оценить будущие изменения в процессах, т.е. интерес представляет не только то, как работают процессы, но и насколько хорошо они работают в терминах измеримых данных (цены, время и так далее). Составление описания того, как работает существующая компания, требует значительного такта и здравого смысла, и ее результаты критичны для успеха последующей работы. Однако не следует слишком углубляться в поиски какой-то неизвестной истины, которая возможно и не требуется.

Должны быть описаны два взгляда (вида) на компанию: внешний и внутренний. Внешний вид строится из перспективы окружения; внутренний вид описывает внутреннюю структуру. Естественно, оба эти вида должны согласовываться. Каждое “что” во внешнем виде должно иметь соответствующее “как” во внутреннем. Внешний вид описывает компанию и ее окружение в терминах прецедентов (см. гл. 5), которые моделируют процессы, взаимодействия с компанией клиентов и партнеров. Интерфейс между каждым прецедентом и его клиентами очень важен и, следовательно, должен быть описан с особой заботой. Модель, основанная на прецедентах, показывает компанию как систему, ее заказчиков - как пользователей этой системой, а ее процессы - как различные прецеденты использования системы клиентами. Внутренний вид описывает внутреннюю модель компании.

4.6

Разработка модели нового бизнеса

В этом разделе данный этап только упоминается для определения его роли в общем контексте. Подробно он будет рассмотрен в гл. 7. Инжиниринг нового бизнеса начинается, когда руководство приняло решение начать процесс реинжиниринга, основанный на спецификации целей компании. Разработка нового бизнеса состоит из четырех основных этапов:

① **Разработка внешнего вида новой компании (“ЧТО” модели).** Детально описываются новые и измененные процессы, особое внимание уделяется их взаимодействию с внешней средой. По сравнению со спецификацией целей компании это описание должно быть более полным и формальным. Оно не должно оставлять сомнений относительно того, что оно означает.

② **Разработка внутренних видов новой компании (“КАК” модели).** Эти модели являются объектно-ориентированными, и каждая из них имеет собственную цель. Можно моделировать каждый процесс в соответствии с тем, какие работы он включает и как они связаны, или с тем, на какие продукты или субпродукты он воздействует. Эти модели могут представлять собой более или менее реальное описание будущей компании, т.е. принимать в расчет ее организацию и географическое расположение. Эти модели также могут быть более абстрактными и описывать идеальную модель компании. Количество моделей, которое вы решите строить, и то, что они будут представлять, - это проектное решение, и оно должно быть сделано в начале работы по реинжинирингу.

③ *Создание информационной системы для поддержки нового бизнеса.* Учитывая важность этой работы, ее описанию будет посвящена гл. 8. Реинжиниринг бизнеса и разработка интегрированной информационной системы - это отдельные, параллельные и сильно связанные работы, т.е. модель бизнеса перепроектированной компании и модель требований к информационной системе должны точно состыковываться друг с другом.

④ *Тестирование перепроектированной компании в небольшом масштабе до ее внедрения.* Результат инжиниринга нового бизнеса - это модель перепроектированной компании.

4.7

Внедрение новых процессов

Как только разработаны и протестированы новые процессы, наступает пора их внедрения в компанию. Это очень ответственный момент, потому что необходимо продолжать работу существующих процессов (в то время как внедряются новые), не нарушая окружение, в котором они работают.

Окружение не должно заметить, что процессы протекают не так, как оно ожидает. Однако может понадобиться изменить процесс таким образом, чтобы клиент получил опыт работы с компанией в новых условиях. В этом случае клиент должен ясно понимать новый процесс и полностью его принимать. В случае только внутренних изменений процесса клиент не должен заметить никакой разницы - новые процессы должны быть совместимы со старыми. Существуют различные способы аккуратного внедрения нового процесса. Как правило, после завершения пилотного проекта выбирают подразделение, специалисты которого имеют наилучшие шансы на достижение успеха при внедрении новых процессов.

4.8

Управление проектом по реинжинирингу бизнеса

4.8.1

Адаптация методологии к выполняемому проекту

В предыдущих подразделах данной главы была кратко описана методология проведения реинжиниринга компании. Более подробно эта методология описана в гл. 6, 7, 8. Однако следует подчеркнуть,

что не существует методологии, полностью подходящей для любого проекта по реинжинирингу, поэтому перед началом разработки приходится проводить адаптацию методологии к особенностям конкретного проекта. Каждый проект предъявляет свои особые требования к методу, например, требования больших проектов (от 10 до 20 исполнителей) отличаются от требований малых (около 5 исполнителей). Метод, хорошо зарекомендовавший себя для малого проекта, далеко не всегда оказывается применимым в случае большого проекта.

Проекты по реинжинирингу могут различаться между собой по следующим аспектам. В первую очередь проект по реинжинирингу зависит от размеров реконструируемой компании и масштаба реинжиниринга (отдел, компания в целом или совокупность компаний в рамках отрасли). Очевидно, что моделирование и тем более реинжиниринг бизнес-процессов большой корпорации (как IBM, Хегох, АВВ) или отрасли представляет собой значительно более сложную задачу, чем реинжиниринг маленькой компании. Разница заключается в количестве сущностей и событий, подлежащих рассмотрению. Кроме того, сложный бизнес, как правило, географически распределен (как в пределах одной страны, так и между странами и континентами), и его инфраструктура оказывается настолько большой и сложной, что едва ли может быть смоделирована детально. Поэтому в случае больших компаний приходится прибегать к специальным средствам работы со сложными системами (см. разд. 5.7).

Важную роль здесь играет степень поддержки и участия высшего руководства компании. Отметим, что для достижения успеха проект должен реализовываться под контролем одного из высших руководителей, глубоко заинтересованного в успешном его осуществлении (руководство при этом должно быть готово пойти на риск).

Степень поддержки со стороны служащих компании не только определяет особенности конкретного проекта, но от нее нередко зависит успех проекта в целом. Предпочтительнее проекты, которые рассматриваются с точки зрения роста и расширения фирмы, а не сокращения размеров и расходов, поскольку первые не вызывают большого сопротивления вводимым новшествам со стороны сотрудников. Особое внимание рекомендуется уделять управляющим среднего звена, поскольку именно их деятельность в наибольшей степени будет изменена в результате реинжиниринга. Таким образом, в работы по проекту входит и решение задач мотивации - разъяснение задач реинжиниринга всем служащим компании, привлечение их на свою сторону и демонстрация преимуществ обновленного бизнеса.

Особенности проекта определяются и опытом разработчиков (как в области реинжиниринга в целом, так и в использовании выбранной ими методологии), а также степенью их знакомства с конкретным бизнесом.

Среди других факторов можно указать также специализацию компании (характер бизнеса, готовность компании к реинжинирингу), особенности имеющихся в компании информационных систем, стиль руководства работ по проекту и др.

Объектно-ориентированный инжиниринг бизнеса предоставляет ряд методик, обеспечивающих стабильное и ровное выполнение проекта. Однако следует выбирать только те из них, которые действительно будут помогать вашей работе и не будут создавать лишних задержек и препятствий. Следовательно, объектно-ориентированный инжиниринг бизнеса должен быть настроен таким образом, чтобы его было легко приспособить к нуждам конкретного проекта.

На основе полного набора методик объектно-ориентированного инжиниринга бизнеса необходимо решить, как будет осуществляться данный проект, наметить направления, в которых он должен двигаться, и определить вид желаемого результата. Чтобы получить наилучший результат, следует заранее провести оценку предполагаемых к использованию объектно-ориентированных моделей инжиниринга, архитектурных решений, документации и этапов работ. Основываясь на полученных оценках, следует принимать решения о том, какие методики метода и в каком объеме будут использовать в данном проекте. Некоторые специалисты называют каждый конкретный способ применения метода *прецедентом разработки* (development case) [10].

Перед началом работы над проектом целесообразно составить документ, определяющий соглашения, которые осуществляют настройку общего метода на данный проект. Этот документ (называемый *описанием прецедента разработки*) должен быть составлен экспертом по методу (см. разд. 2.6) совместно с лидером проекта и, возможно, экспертом по реинжинирингу. Но если в этой работе будет принимать участие слишком много сотрудников, то составление документа может неоправданно затянуться. Чтобы сэкономить время, перед началом проекта распространите среди всех исполнителей проекта информацию о принятых соглашениях. Соответствующий документ обычно описывает отклонения общего метода в следующих аспектах:

- модели;
- архитектурные решения;
- перечень промежуточных и окончательных документов, которые следует разработать;
- оглавление для каждого документа;
- конкретные виды работ по проекту.

Решите, какие объектно-ориентированные модели следует использовать в ходе реинжиниринга бизнеса: например, нужна ли идеальная объектная модель, реальная объектная модель или их сочетание? В гл. 7 приведен критерий, использование которого облегчает решение этого вопроса в случае конкретного проекта.

Для каждой модели определите основные архитектурные концепции. Многие из решений, принимаемых на этом шаге, получаются в результате взвешивания всех “за” и “против”. Например, в случае использования отношения “состоит из” между объектами модель выигрывает в ясности и простоте, но при этом разработчикам придется потратить больше времени на освоение этого подхода.

Определите, какие документы следует составить, и составьте оглавление для каждого из них. Можно сэкономить много времени и сил, если составить образцы выпускаемых документов. В образце указывается, какого рода информация должна содержаться в каждом разделе и с какой степенью подробности она должна быть изложена. Составление образцов способствует единообразию выпускаемой документации, что особенно важно в тех случаях, когда документы подготавливаются разными исполнителями.

Следует адаптировать работы, предлагаемые методом, чтобы обеспечить оптимальное достижение конкретного результата. Обычно это включает отсеивание работ и документов, ненужных для вашего проекта.

4.8.2

Организация проекта и управление им

Организация и управление проектом по реинжинирингу требуют глубокого понимания проблем и “подводных камней”. Необходимым, но не достаточным условием для успешного реинжиниринга является хорошее управление проектом. Теме управления проектом по реинжинирингу посвящено много литературы, поэтому сосредоточимся на особенностях применения объектных методов в управлении проектами по реинжинирингу и на том, что следует принимать во внимание при использовании этой новой технологии в работе над конкретным проектом.

Можно выделить следующие работы по управлению проектом [10]:

- ① *предварительное изучение*: включает постановку задачи проектирования, оценку разнообразных целей, потребностей и идей, обеспечивающих техническую и экономическую осуществимость проекта;
- ② *планирование*: включает организацию и планирование работ по проекту (в том числе - составление детальных планов со сроками исполнения и с указанием требуемых ресурсов);
- ③ *выполнение*: собственно выполнение проекта в соответствии с составленными ранее планами;

4 окончание: работы, выполняемые по завершении проекта и состоящие в резюмировании всех предложений по улучшению самого проекта и методов его выполнения.

Объектно-ориентированный инжиниринг бизнеса используется в основном во время этапов предварительного изучения и выполнения. Разработанные модели позволяют направлять проектные работы. Обратный инжиниринг и разработка образа будущей компании выполняются во время предварительного изучения. (Если реинжиниринг требует существенных усилий или высокого уровня подготовки в области новых методик, можно включить работу по обратному инжинирингу как отдельную работу в проекте).

Прямой инжиниринг и внедрение выполняются в течение этапа выполнения. Соответствие между работами по реинжинирингу и работами по управлению проектом показано на рис. 4.4.

Приведенное нами идеализированное соответствие чем-то напоминает ранние “водопадные” модели, поскольку в нем точно так же зафиксирована последовательность работ, и выполнение очередного этапа становится возможным только по завершении предыдущего. Получается, что этап визуализации (образ будущего) должен быть выполнен, пересмотрен и заморожен до начала работ по прямому инжинирингу.

Следует понимать, что, работая над одним этапом, вы наверняка обнаружите, что результаты предыдущего этапа вас не совсем устраивают и необходимо кое-что переделать. Несмотря на то, что основной поток работ имеет фиксированное “русло” - начинается с образа будущего, проходит через обратный инжиниринг и заканчивается прямым, - работа может итеративно возвращаться к предшествующим этапам в рамках каждого отдельного этапа.

В дополнение к этим итерациям следует учитывать влияние инкрементного характера разработки (см. гл. 6). Каждый инкрементный шаг удобно рассматривать как самостоятельный проект, и тогда можно естественным образом представлять измененные модели как очередные версии.

Как правило, работа начинается с этапа визуализации, и после выполнения некоторых начальных процедур инициируется параллельное выполнение этапа обратного инжиниринга. Возможно несколько итераций между этими двумя этапами. После того как будет сформирована адекватная картина существующего бизнеса, инициируется этап прямого инжиниринга. После нескольких итераций этап визуализации заканчивается, и его результатом оказывается первая версия спецификаций целей. Прямой инжиниринг исходит из этих спецификаций, а также моделей, построенных в ходе обратного инжиниринга.

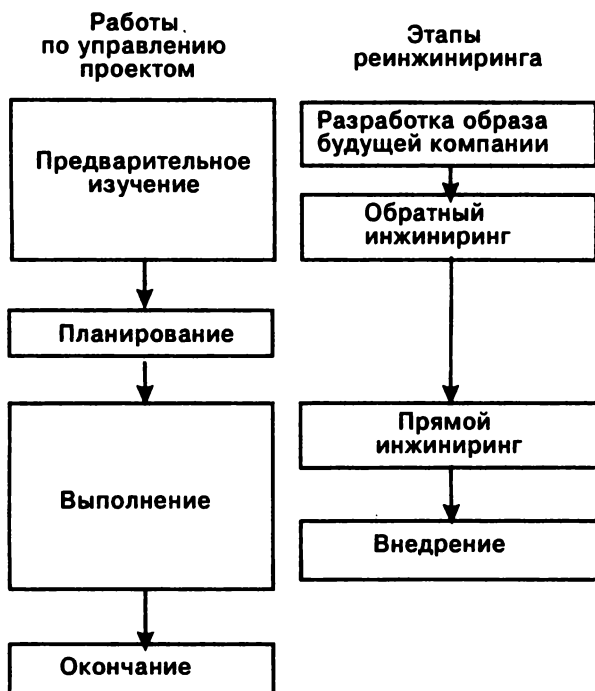


Рис. 4.4. Соответствие между этапами реинжиниринга и работами по управлению проектом

С учетом инкрементной разработки картина еще усложняется. В ходе выполнения первой версии, включающей этапы визуализации, прямого и обратного инжиниринга, инициируется вторая версия, так что некоторое время она выполняется параллельно с первой. В результате каждого этапа визуализации формируется очередная версия спецификаций целей. Эта спецификация, содержащая список целей системы в терминах бизнес-процессов (а также, возможно, дополнительных целей, учитывающих сроки поставок, ресурсы и качество), представляется на одобрение высшего руководства корпорации. Различие между спецификациями версии 1, 2 и т.д. заключается как в уточнении ранее сформулированных целей, так и в добавлении новых.

Наиболее серьезная проблема при формулировании спецификации задач и целей состоит в том, что эти спецификации должны быть понятны высшему руководству компании и в то же время содержать достаточно полные и четкие указания для разработчиков. Сущест-

вуют три серьезные причины для изменения спецификаций целей в ходе выполнения работ:

- цели никогда не удастся сформулировать окончательно ясно, поэтому в ходе выполнения конкретной работы происходит их постоянное уточнение;
- исполнители проекта обнаруживают, что для достижения всех целей им недостаточно времени или имеющихся ресурсов, поэтому они вынуждены отказаться от некоторых целей;
- некоторые цели, запланированные в начале работы над проектом, оказываются нереалистичными и не могут быть достигнуты.

Изменение или добавление новых целей неизбежно увеличивает стоимость проекта и может потребовать увеличения сроков разработки. Однако лучше проводить необходимые изменения как можно раньше, ближе к началу цикла разработки. Цель, отработка которой на этапе прогнозирования оценивается в 200 чел.-час. (из ресурсов команды по реинжинирингу), может потребовать 1000 чел.-час., если она будет введена на этапе прямого инжиниринга.

4.8.3

Виды обсуждения проекта

Обсуждения (reviews) - это основное средство обеспечения должного качества (при работе над проектом). В этом разделе кратко остановимся на способах проведения обсуждений при инжиниринге бизнеса. Начнем с терминологии: будем различать формальные и неформальные обсуждения. Цель *формального обсуждения* состоит в принятии решения о возможности перехода к очередному этапу разработки. Обсуждения этого типа проводятся по достижении каждого этапа проекта, причем они проводятся в расширенном составе - с привлечением клиентов или заказчиков.

Цель *неформального обсуждения* состоит в выявлении ошибок. Эти обсуждения могут проводиться в любое время разработки, например, по завершении какой-либо локальной работы для проверки ее результатов, перед продолжением процесса разработки. В неформальных обсуждениях, как правило, число участников ограничено - несколько разработчиков и, возможно, представитель группы поддержки качества.

Когда и где использовать различные виды обсуждений, зависит от масштабов проекта. В случае малых и средних проектов формальное обсуждение обычно организуется по завершении каждого основного этапа, то есть после того, как построена первая версия какой-либо модели. Неформальные обсуждения могут проводиться после каждого вида деятельности внутри этапа.

IEEE [4] определяет различные виды обсуждений в своем глоссарии стандартов. Рассмотрим три вида обсуждений, представляющих собой трансформацию стандартов IEEE на случай объектно-ориентированного инжиниринга бизнеса.

Обсуждение: совещание, проводимое для проверки результатов и принятия решений по ним. Обсуждения нужны для того, чтобы убедиться в полноте и согласованности всех полученных к текущему моменту результатов. Это единственный способ оценки результатов, в котором следует принимать участие высшему руководству. Если оно будет участвовать в других совещаниях рабочей группы, то оценка результатов и обсуждение документа, вероятнее всего, перейдет в разбирательство над командой реинжиниринга. Во время обсуждения рассматривается общий результат проекта по реинжинирингу и принимается решение о переходе к следующему этапу разработки. Такое решение должно приниматься высшим руководством компании. При проведении обсуждений статус каждого документа контролируется: выясняется, какие типы оценок применялись и сколько серьезных ошибок было выявлено.

В обсуждениях участвуют лидеры проекта, члены команды по реинжинирингу и владельцы процессов. Они изучают все инспекционные записи, сделанные во время обсуждаемого этапа разработки, и созданные на этом этапе документы. Если выясняется, что обнаруженные нерешенные проблемы оказывают слишком большое влияние на последующую деятельность, то прежде чем переходить к ней, следует внести необходимые коррективы.

Инспекция: наиболее мощный из всех методов оценки. Представляет собой формальную методику оценки, результаты которой должны быть зафиксированы письменно. Инспекции обеспечивают хороший контроль за выполнением проекта по реинжинирингу и за статусом результатов. Они могут также давать ценную информацию на будущее. При проведении инспекций выделяют следующие конкретные роли ее участников:

арбитр - председательствующий на совещании; обеспечивает создание творческой и продуктивной атмосферы и направляет деятельность участников на выявление ошибок;

секретарь - фиксирует каждую обнаруженную ошибку, ее приоритет и степень серьезности;

инспектор - несет ответственность за критический разбор и комментирование ошибок (или возможных ошибок);

автор, или проектировщик, - дает необходимые пояснения.

Прежде чем арбитр объявит совещание законченным, необходимо решить, как поступить с обсуждавшимся документом: принять без изменений, принять после исправления выявленных ошибок или не принимать (в последнем случае он должен быть доработан и представлен для проведения очередной инспекции). Инспекционное со-

вещание должно длиться не более двух часов, и число участников не должно превышать восьми человек.

Прогон (Walkthrough): метод оценки, при котором проектировщик “проводит” одного или нескольких членов команды реинжиниринга, участвующих в проекте, через разработанный ими сегмент модели. Другие члены команды делают замечания по стилю, методам, возможным ошибкам, нарушению стандартов реинжиниринга и т.д. Проектировщик начинает с пояснений по общему содержанию документа, после чего переходит к деталям, прочитывая и объясняя каждую главу. Все ошибки, обнаруженные во время прогона, должны быть зафиксированы проектировщиком для дальнейшей проработки. Однако никаких формальных записей и протоколов не требуется: цель прогона - дать проектировщику информацию, необходимую для коррекции его работы. От участников не требуется специальной подготовки к совещанию, проведение которой означало бы, что у них есть проблемы с обнаружением ошибок. Напротив, совещание представляет собой хорошее упражнение с широким обменом информацией.

Обсуждения и инспекции можно считать формальными, а прогоны - неформальными способами оценки. Хотя каждое обсуждение уникально и посвящается конкретной модели или даже объекту, их объединяют некоторые общие особенности. Не претендуя на исчерпывающий перечень того, что должно обсуждаться в моделях инжиниринга бизнеса, приведем некоторые примеры.

Все виды обсуждений направлены на проверку моделей с точки зрения их полноты, избыточности, структуры, ясности и понятности, отслеживания версий документов, стандартов и т.д. Как именно осуществляется проверка, в значительной степени зависит от цели обсуждения.

Фокус внимания при обсуждении зависит от вида обсуждаемой модели. Например, при обсуждении П-модели особое внимание уделяется следующим вопросам:

- Правильно ли определены границы бизнеса?
- Соответствуют ли прецеденты спецификациям целей?
- Нужно ли изменять спецификацию целей?
- Можно ли понять входящие в модель прецеденты?
- Все ли роли из окружения бизнеса описаны в виде субъектов?
- Для всех ли субъектов определены адекватные множества прецедентов?
- Удовлетворительно ли описаны интерфейсы?
- Являются ли потоки событий в прецедентах корректными и полными?
- Все ли альтернативные потоки описаны?

Разумеется, реальные вопросы зависят от конкретной модели, но общие намерения участников обсуждения, по-видимому, достаточно ясны. При обсуждении других моделей намерения обычно такие же:

найти ошибки, допущенные в ходе процесса разработки, и обеспечить высокое качество продукта.

При проведении обсуждений можно использовать различные методики и приемы. Для обеспечения высокого качества разрабатываемых систем необходим систематический подход. Соответствующие методики и приемы подробно рассмотрены в [7], [8] и [9].

В проведении обсуждений важная роль отводится менеджерам проекта. Важно, чтобы они были согласны с используемой методологией (в нашем случае - объектно-ориентированным инжинирингом бизнеса) и полученными результатами, а кроме того со временем, выделенным на проведение обсуждений. Цифра 5 % от общего времени разработки представляется вполне разумной. Также имеет большое значение подбор компетентных сотрудников для участия в обсуждениях, что должно гарантировать высокое качество продукта и достаточную степень доверия со стороны разработчиков к тем, кто будет обсуждать их работу.

Чтобы добиться хорошей дисциплины и высокого уровня понимания, необходимо привлечь независимую группу поддержки качества. Эта группа может проводить обсуждения по поводу того, насколько последовательно выбранная методология применяется в разработке, а также оценивать различные возможности достижения целей проекта и оценивать потенциальный риск. Но группа поддержки качества не должна играть роль "жандарма"; ей следует в сотрудничестве с командой по реинжинирингу повышать качество того, что делается в ходе разработки.

Итак, ключом к хорошему качеству продукта является тщательное обсуждение работы. Обсуждения требуют времени, но при правильном проведении они оказываются жизненно необходимыми.

Для участников обсуждений очень важно использовать прекрасную возможность более глубоко понять тот материал, который они обязаны прочитать. Настоятельно рекомендуется всем участникам ознакомиться с применяемой методологией до того, как начнется процесс обсуждения.

ЛИТЕРАТУРА

1. Carlzon J. and Lagerstrom T. *Riv Pyramiderna! En bok om den nya manniskan, chefen och leedaren*, 1985.
2. Davenport T.H. *Business Innovation, Reengineering Work through Information Technology*. - Boston: Harvard Business School Press, 1993.
3. Hammer M. and Champy J. *Reengineering the Corporation: A Manifesto for Business Revolution*. - N-Y: HarperCollins, 1993.
4. IEEE STD 729-1983. *Standard Glossary of Software Engineering Terminology*. N-Y: IEEE, 1983.

5. Jacobson I., Christerson M., Jonsson P. and Overgaard G. **Object-oriented Software Engineering - A Use Case Driven Approach**. Reading, MA: Addison-Wesley; N-Y: ACM Press, 1992.
6. Johansson H., McHugh P., Pendlebury J. and Weeler III W. **Business Process Reengineering. Breeakpoint Strategies for Market Dominance**. Chichester: John Wiley & Sons, 1993.
7. Myers G.J. **Software Reliability: Principles and Practices**. N-Y: John Wiley & Sons, 1987.
8. Weinberg G.M. and Freedman D.P. **Handbook of Walkthroughs, Inspections, and Technical Reviews**. - Boston: Little Brown Computer Systems, 1982.
9. Yourdon E. **Structured Walkthroughs**. 4th edn. Englewood Cliffs, N-Y: Prentice-Hall, Yourdon Press, 1989.
10. Jacobson I., Ericsson M., Jacobson A. **The Object Advantage: Business Process Reengineering with Object Technology** //ACM Press. - Addison-Wesley Publishing, N-Y: 1995.

[5]

АРХИТЕКТУРА ИНЖИНИРИНГА БИЗНЕСА

5.1

Требования к бизнес-модели

В интересующем нас контексте слово “архитектура” относится, во-первых, к архитектуре конкретного бизнеса, во-вторых, - к различным типам архитектур, которые могут быть представлены в выбранном языке моделирования. Именно второе значение этого слова будет рассматриваться в данной главе в первую очередь.

Под “архитектурой” компании подразумевают наиболее важные статические или “долгоживущие” структуры компании. Типичные элементы структуры - это функции компании, т. е. отделения, отделы и т. д. Элементами структуры являются также процессы и продукция на различных уровнях, а также ресурсы, как человеческие, так и технические. Одни элементы связываются с другими, вместе они формируют структуру. Каждый элемент имеет своего владельца - некоторого сотрудника компании. Элементы вещественны, осязаемы, им можно присвоить значение (иногда несколько значений), и у них есть диапазоны значений.

Хотя именно архитектура составляет основу модели бизнеса, в модели необходимо отразить и динамику, т. е. то, что компания делает в различных ситуациях. Процессы, которыми следует управлять, и решения, которые должны приниматься в различных ситуациях, очень важны для понимания задач компании.

Попытаемся определить, что надо уметь выразить в бизнес-модели. Для этого уточним такие понятия, как “процесс”, “функция” (“подбизнес”).

Термин “процесс” употребляется в разных контекстах. В моделировании бизнеса он показывает, как клиент может использовать бизнес, и включает в себя полный поток событий в системе, описывающий, как клиент начинает, ведет и завершает бизнес-процесс.

Т. Давенпорт [1] определяет *процесс* как “структурированный, измеряемый набор действий, созданный, чтобы произвести определенный выход для конкретного клиента или рынка”. Это определение наиболее точно отражает интересы тех, кто наблюдает бизнес снаружи, например, для акционеров и клиентов компании.

Термин “процесс” используется и другим образом. М. Хаммер и Дж. Чампи [2] применяют тот же термин для описания в н у т р е н - н е й (а не внешней) деятельности компании. Подобные процессы не имеют внешнего клиента; их результат используется другими внутренними процессами. Примерами таких процессов служат различные процессы управления, а также процессы инфраструктурного характера.

Первое определение соответствует бизнес-процессам (внешним процессам) компании; второе относится к тому, как эти процессы подразделяются на внутренние процессы. По существу, с этими процессами желательно обращаться по-разному и, следовательно, необходимо уметь их различать.

Язык моделирования должен описывать различные типы задач или внутренних процессов, из которых состоит почти каждый бизнес-процесс, а также способ взаимодействия этих внутренних процессов при обслуживании клиента.

Другое важное понятие, которое надо обсудить, - это термин “функция”, или “подбизнес” (sub-business) [5]. Большой бизнес целесообразно разделить на несколько функций (подбизнесов), каждая из которых покрывает конкретную область компетенции. Заметим, что в процессно-ориентированной компании *подбизнес* - это не традиционная область ответственности, такая, как финансы или маркетинг, а область компетенции, такая, как знание конкретного типа продукции, производимой компанией. Подбизнес не имеет менеджера, но имеет владельца ресурса, который “совершенствует” персонал и “продает” его руководителям (лидерам) процессов. Лидеры процессов имеют предусмотренные в бюджете ресурсы, на которые они могут “покупать” требуемый персонал. Проще говоря, используя владельцев процессов и владельцев ресурсов, традиционная ответственность разделяется на два компонента: один - ведет компанию к ее финансовым целям, другой - совершенствует ресурсы персонала, способствуя достижению финансовых целей.

Язык моделирования должен выражать, как внутренний процесс реализуется при помощи человеческих или технических ресурсов и из каких функций эти ресурсы будут взяты. Особенно важно уметь показать, как процесс может поддерживаться информационной системой. Должна существовать согласованность между внутренним процессом в организации бизнеса и требованиями, предъявляемыми к поддерживающей информационной системе (см. гл. 8).

Понятия “товар” и “услуга” или их обобщение - понятие “продукция”, также необходимо уметь моделировать. Продукция может состоять из других продуктов. Она имеет характеристики, которые описывают, что с ней можно делать и что она содержит. Строительный материал для описания продукции - это объект с его операциями и атрибутами.

Кроме того, необходимо уметь описывать потоки событий при выполнении бизнес-процесса. Необходимо описывать входные данные из внешнего мира; действия (операции) процесса, которые он производит над исходными данными; потребляемые ресурсы; решения, которые будут приняты, и выход (результат), который процесс отправит во внешний мир. Так как “вход” и “выход” - это способы общения процесса с клиентом, они должны быть ориентированы на клиента.

Наконец, существуют общие требования к моделированию бизнеса в целом. При конструировании компании обязательно необходимо иметь в виду, *что компания будет подвержена изменениям*. Следовательно важно, чтобы язык был гибким. Должна существовать возможность менять процессы, но при этом производить продукцию для клиентов бизнеса. Необходимо также иметь возможность адаптировать процессы к новым ситуациям. Например, если компания открывает филиалы, должна быть предусмотрена возможность специализировать процесс продаж так, чтобы он работал и в филиале.

При разработке модели бизнеса в ходе прямого или обратного реинжиниринга необходимо создать две модели компании: внешнюю и внутреннюю (см. гл. 6,7). *Внешняя модель* описывает компанию и внешний для нее мир. Она описывает процессы, которые удовлетворяют интересы клиентов и интересы вне компании. Интерфейс между каждым процессом и окружающей средой жизненно важен и должен быть описан с особым вниманием.

Внутренняя модель компании описывает: как строится каждый бизнес-процесс из различных рабочих задач (внутренних процессов) и какие ресурсы он использует. Внутренняя модель использует объекты, соответствующие рабочим задачам, и объекты, соответствующие предметам бизнеса, например продукция.

Большинство специалистов по БПР прямо или косвенно утверждают, что для описания внешней и внутренней моделей достаточно статического языка моделирования. Например, Дж.Мартин [6] для описания моделей рекомендует использовать OMW (см. разд. 9.1) - статическое ИС фирмы IntelliCorp, объединяющее статическое объектно-ориентированное CASE-средство со статической экспертной системой. Аналогично И.Якобсон [5], указав, что модели являются отчасти динамическими, для описания и внешней, и внутренней модели использует по сути статический язык с зачатками динамики.

С точки зрения авторов данной книги, мнение этих ведущих специалистов отражает не суть явления (а суть в том, что обе модели в большинстве случаев являются динамическими), а возможности используемых ими инструментальных средств. Дело в том, что до недавнего времени для моделирования деятельности компании разработчики могли использовать только статические ИС, так как других средств не было. Типичными ИС, применяемыми для моделирования деятельности компаний, являлись разнообразные CASE-средства, которые описывали деятельность компании в статических понятиях, например, E/R-диаграммы, диаграммы потоков данных и т.п. (см. разд. 3.4). Эти средства позволяли описать компанию в виде взаимосвязанной совокупности блоков (отделений, отделов, лабораторий, групп и т.п.), указывая при этом их функции, входы, выходы, диапазоны передаваемых параметров и т.п. Однако эти ИС не позволяли описывать реальное, текущее состояние компании и ее подразделений, изменяемое во времени. В последнее время в ряд CASE-средств начинают вводить компоненты, позволяющие в ограниченном виде описывать некоторые аспекты динамики. Не вызывает сомнения, что любая модель компании (в том числе и статическая) лучше, чем отсутствие модели, однако для многих компаний их деятельность не может быть адекватно описана без привлечения динамики, т.е. без учета временного фактора. Например, в Приложении 2 описана модель автоматизированного грузового комплекса (Шереметьево-Карго), обеспечивающего прием (отправку) и хранение грузов при авиационных перевозках. Учитывая высокую динамичность подобных компаний (самолеты прибывают или уходят каждые несколько минут), адекватно описать их модель статическими средствами невозможно даже теоретически. Другим примером компаний, требующих для их описания динамической модели, являются любые компании, использующие для производства продукции технологические процессы (металлургические, химические, перерабатывающие и др.).

В отличие от статических ИС, динамические ИС позволяют:

- динамически (т.е. во время работы приложения) создавать, модифицировать и удалять объекты;
- динамически устанавливать отношения между объектами;
- хранить историю изменения значений переменных во времени;
- динамически обновлять значение переменных (опрашивая внешнее окружение или выполняя умозаключения), если значения переменных “устарели” (см. Приложение 1);
- выполнять умозаключения, учитывающие не только значения переменных, но и заданные моменты времени;
- динамически назначать и освобождать ресурсы (например, грузчиков, трактора, тележки - см. Приложение 2), используемые в различных операциях процесса;

- динамически вычислять показатели стоимостей и производительности отдельных операций и всего процесса в целом;
- накапливать требуемые текущие статистические показатели в любой точке моделируемого процесса;
- изменять модель, не прерывая процесс ее функционирования.

Итак, в общем случае и внешняя, и внутренняя модель являются динамическими, поэтому при моделировании компании следует применять развитые ИС, позволяющие описывать динамику. Анализ ИС приведен в разд. 9.2, а в разд. 9.3 описано динамическое интегрированное ИС - ReThink (фирма Gensym), построенное на базе оболочки для создания экспертных систем G2 фирмы Gensym (см. Приложение 1). Тем не менее учитывая то, что большинство ИС ориентированы только на представление статистики, в данной главе для описания моделей использованы в основном статические языковые средства, а динамические возможности представлены в разд. 9.3 (ReThink), Приложение 1 (описание G2) и Приложение 2 (описание динамической модели Шереметьево-Карго средствами ReThink).

5.2

Внешняя модель

Итак, внешняя модель описывает бизнес и его окружение. Бизнес состоит из всех относящихся к делу бизнес-процессов, а внешнее окружение состоит, например, из клиентов, партнеров и поставщиков, которые принимают участие в этих процессах. В предлагаемом подходе процессы моделируются при помощи прецедентов, а окружение моделируется при помощи так называемых действующих лиц (будем для краткости называть их *субъектами*). Понятие *прецедент* (см. подразд. 5.2.3) представляет собой некоторый способ использования бизнеса клиентом, т.е. некоторый процесс. Внешнюю модель будем называть *П-моделью (прецедент-моделью)*.

П-модель также показывает, как внешнее окружение взаимодействует с бизнесом, т.е. как отдельные субъекты общаются с бизнесом посредством прецедентов. П-модель описывает бизнес так, как он виден извне, т.е. так, как он воспринимается теми, кто хочет его использовать. Следовательно, структуры внутри бизнеса, которые невидимы для субъектов, не следует описывать в П-модели.

Ниже будут уточнены следующие моделируемые понятия: *бизнес-система*, *субъект* и *прецедент*. Затем будет показано различие между *классами* и *экземплярами классов* для прецедентов и субъектов и описаны некоторые виды отношений между прецедентами.

5.2.1

Бизнес-система

Понятие, которое в разрабатываемых моделях будет символизировать бизнес, - это бизнес-система (для краткости система). Важно определить границу между системой и ее окружением. Когда работа по моделированию только начинается, эта граница не определена, но когда модель полностью построена, граница должна быть ясно очерчена. Граница системы не очевидна и может меняться при различных рассмотрении бизнеса.

Для обозначения некоторой бизнес-системы в П-модели будем использовать прямоугольник с расположенным над ним именем этого бизнеса (рис. 5.1). Все, что внутри прямоугольника, относится к бизнес-системе, а все, что вне его, - к ее внешнему окружению.

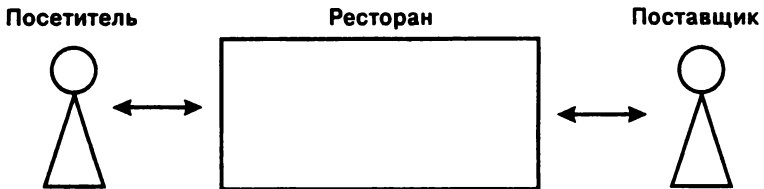


Рис.5.1. Бизнес-система Ресторан и ее окружение

5.2.2

Субъекты (действующие лица)

Чтобы создать корректную модель бизнеса, важно очертить его окружение. Например, в окружении присутствуют клиенты, и именно они накладывают на бизнес наиболее существенные требования.

Субъект обозначает роль, которую кто-то или что-то может играть по отношению к бизнесу. В рассматриваемых моделях окружение представлено субъектами. Субъекты обозначают все то в окружении, что взаимодействует с бизнесом и подлежит моделированию. Например, это клиенты, поставщики или партнеры. Субъектов можно разделить на две группы: человеческие и технические. Например, компьютерная система другой компании может быть представлена как технический субъект. С другой стороны, работников ресторана или его технику не следует рассматривать как субъекты, так как они являются неотъемлемой частью самого бизнеса, т. е. они представляют собой ресурсы, которые используются для выполнения задач внутри системы.

На рис. 5.1 изображена бизнес-система Ресторан, субъекты показаны в виде графических фигурок. Стрелки отношений, нарисованные между субъектами и бизнес-системой, означают, что они могут общаться или кооперироваться между собой. Важно понимать, что субъект представляет собой абстракцию кого-либо или чего-либо, использующего бизнес. Субъект может представлять различные виды явлений в окружении компании. В нашем примере это Посетитель и Поставщик.

Не следует путать реальных людей с субъектами. Реальная личность, в отличие от субъекта, может играть несколько ролей в бизнес-системе: конкретный человек может быть и посетителем, и одним из поставщиков ресторана.

5.2.3

Прецеденты

***Прецедент** - это последовательность транзакций в системе, выполняемых для получения измеримой потребительской ценности для некоторого индивидуального субъекта бизнес-системы.*

Рассмотрим более подробно ключевые слова из этого определения:

Прецедент. Приведенное выше определение фактически говорит о конкретном потоке событий через систему, т. е. об экземплярах событий, а не о классе событий. Существует большое количество вариантов хода событий, некоторые из них очень похожи, поэтому чтобы сделать П-модель более выразительной, целесообразно группировать подобные варианты хода событий и называть каждую группу классом прецедентов. Например, конкретный прецедент может зависеть от того, новый ли клиент, где он находится, будет ли завтра выходной и так далее. Все эти альтернативы удобно сгруппировать в один прецедент, называемый, например, Продажа. Если не группировать конкретные прецеденты, то получится огромное количество вариантов и модель станет очень малопонятной. Итак, когда говорят, что идентифицируют и описывают прецедент, то имеют в виду, что идентифицируют и описывают класс.

Индивидуальный субъект. Это понятие упрощает нахождение правильных прецедентов, т.е. позволяет избегать слишком сложных прецедентов. Рассмотрение прецедента важно начинать с индивидуальных субъектов - с экземпляров действующих лиц. Например, при моделировании компании, специализирующейся на продаже некоторой продукции, необходимо осознавать, что субъект, названный "клиент", на самом деле представляет трех различных клиентов: во-первых, *простого потребителя* продукции (каких много); во-вторых, *покупателя продукции*, т. е. кого-то, кто разбирается в закупках, но

не обязательно знает, для чего используется продукция; и, в-третьих, того, кто может компетентно судить о продукции в сравнении с конкурирующими продуктами. Каждый из этих случаев требует отдельного прецедента, так как субъекты играют в них различные роли по отношению к системе.

Измеримая потребительская ценность. Эти слова служат ключом для выбора не слишком детального уровня прецедента. Должна быть предусмотрена возможность оценить эффективность прецедента в терминах цены или стоимости. Например, обращение в банк за кредитом имеет ценность для клиента банка.

Транзакция - это неделимое множество действий, которые или выполняются все целиком, или не выполняются вообще. Она запускается при получении системой стимула от субъекта или при наступлении определенного момента времени. Транзакция состоит из набора действий, решений и передачи стимулов некоторому субъекту (субъектам).

Графически прецедент будем представлять в виде окружности. Для именования прецедентов, описывающих некоторые процессы, целесообразно использовать отглагольные формы, чтобы не путать процессы с объектами. На рис. 5.2 приведены следующие прецеденты:

Обслуживание Обеда или Ужина - посетитель приходит в ресторан и его обслуживают. Посетитель может выбрать блюдо из разнообразного меню или даже заказать блюдо по своему рецепту.

Обслуживание Завтрака - посетитель приходит в ресторан и его обслуживают в процессе Завтрака. Он может выбрать блюдо из ограниченного меню.

Обслуживание Мероприятия - посетитель заранее заказывает обслуживание торжественного Мероприятия (свадьба, день рождения и т.п.) для группы приглашенных им людей. В этом случае могут быть заказаны блюда, отсутствующие в ресторанном меню. Кроме того, может быть заказан оркестр и исполнители.

Поставка Продуктов - поставляются компоненты, необходимые для приготовления блюд.

Когда окружение бизнеса идентифицировано, можно начинать идентифицировать бизнес-систему изнутри. Как правило, эта работа выполняется итеративно. Бизнес-система Ресторан имеет окружение в виде двух типов субъектов: Посетителя и Поставщика. Ресторан взаимодействует с окружением посредством четырех прецедентов (см. рис. 5.2): Обслуживание Завтрака, Обслуживание Обеда или Ужина, Обслуживание Мероприятия и Поставка Продуктов. Конкретные Посетители и Поставщики могут использовать эти четыре прецедента.

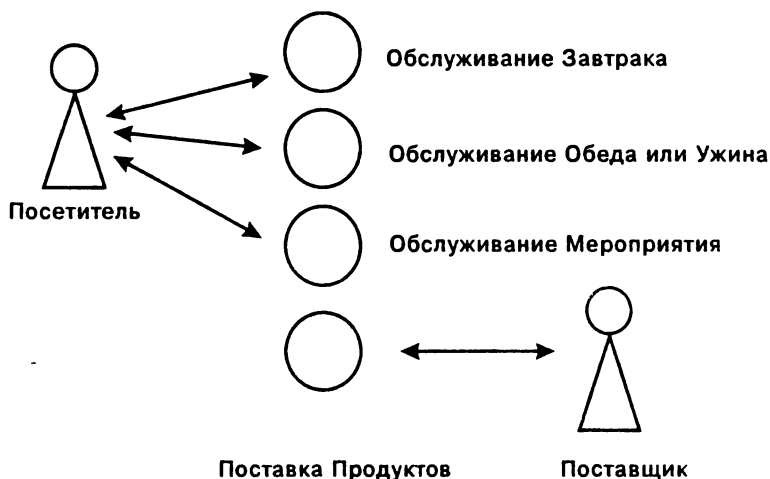


Рис.5.2. Прецеденты бизнес-системы Ресторан

5.2.4

Класс и экземпляр

При описании модели нужно иметь возможность говорить как об общих характеристиках некоторого типа субъектов, так и об уникальных характеристиках конкретного субъекта. Другими словами, надо иметь возможность различать класс субъектов и фактических субъектов. В модели реальные люди являются не экземплярами некоторого класса, а реализациями экземпляров этого класса. Субъект представляет собой роль, которую реальные люди могут играть. Один человек может играть несколько ролей, т. е. реализовывать несколько экземпляров субъекта.

Таким же образом следует различать класс и экземпляр прецедента. Класс прецедентов определяется его описанием, а экземпляр прецедента - действиями, которые выполняются, когда поток событий, соответствующий описанию прецедента, проходит через систему.

Класс прецедентов может содержать несколько альтернативных путей через систему, но экземпляр следует только одному из них.

Важно уметь различать экземпляры и классы, так как необходимо:

- уметь моделировать несколько параллельных потоков событий в бизнесе. При этом несколько экземпляров одного класса прецедента могут выполняться параллельно;

- иметь возможность идентифицировать характеристики, общие для похожих классов, которые не требуется описывать в конкретных экземплярах. Экземпляр руководствуется описанием класса и может общаться с экземплярами субъектов;

- удостовериться, что конкретные клиенты (экземпляры субъектов) связаны с одним (и только с одним) экземпляром данного прецедента.

5.2.5

Описание прецедента

Чтобы придать П-модели большую глубину, составляются детальные описания каждого прецедента. Следует не просто описывать поток прецедентов, но и то, как они взаимодействуют с окружением, т. е. с субъектами.

Пример. Рассмотрим более подробно пример “Обслуживание Обедов или Ужины”. Когда посетитель входит в ресторан, его встречают у дверей и предлагают снять пальто. После этого его усаживают за столик и дают меню. Когда субъект Посетитель обдумал свой выбор, официант предлагает ему сделать заказ; субъект может сам привлечь внимание официанта для выполнения заказа. Посетитель делает заказ официанту, заказ передается на кухню, где готовятся соответствующие блюда. Когда заказ готов, его подают посетителю. По окончании еды ожидается, что посетитель даст сигнал официанту и заплатит. Заплатив, он берет свое пальто в гардеробе и покидает ресторан. На этом прецедент завершается.

Это описание упрощено по сравнению с тем, как должен быть описан прецедент для П-модели. Дело в том, что в данном прецеденте возможны различные альтернативы в ходе событий, такие как: “какое-либо блюдо не готово” или “посетитель не может заплатить”. Если попытаться описать прецедент, содержащий много альтернатив хода событий, то он станет непонятным. Следовательно, было бы целесообразно использовать некоторую форму структурного описания. Тогда пример выглядел бы следующим образом:

Основной поток событий:

А. Прецедент начинается, когда Посетитель входит в ресторан.

Б. Посетитель имеет возможность оставить пальто в гардеробе, после этого он приглашается за столик и получает меню.

В. Когда Посетитель ознакомился с меню, официант просит его сделать заказ. Посетитель может сам позвать официанта и сделать заказ.

Г. Когда Посетитель сделал заказ, кухня получает информацию о продуктах и напитках для его приготовления.

Д. На кухне некоторые основные компоненты, такие, как соусы, рис и картофель, уже приготовлены. Таким образом, приготовление состоит в объединении этих основных компонентов вместе, добавлении специй и т.п. и определении того, что следует сделать непосредственно перед тем, как подать блюдо. Кроме того, из холодильника достают различные напитки.

Е. Когда блюдо готово, его подают Посетителю. Когда обед завершен, ожидается, что посетитель даст сигнал Официанту и заплатит.

Ж. Когда оплата произведена, Посетитель может взять свое пальто в гардеробе и покинуть ресторан. На этом прецедент завершается.

А л ь т е р н а т и в ы х о д а с о б ы т и й :

- Альтернатива А: если ресторан переполнен, Посетитель имеет выбор посидеть в баре, пока освободится столик, или уйти из ресторана. В первом случае прецедент используется с этапа Б после некоторого ожидания. Если субъект уходит, пример завершается.

- Альтернатива В: если на этапе В выясняется, что блюдо, которое хочет Посетитель, сегодня не подается, то официант предлагает замену блюда. Если Посетитель решает заказать что-либо другое, то прецедент продолжается с этапа Г, иначе посещение ресторана завершается.

Рассматривая прецеденты как машину состояний-событий, легче понять, как находить прецеденты и как их следует описывать. Другими словами, прецедент можно рассматривать как машину состояний-событий, в которой состояние экземпляра прецедента представляет, какие стимулы могут быть получены следующими и какие стимулы могут вызвать переход в другое состояние, после завершения транзакции. Экземпляр прецедента, будучи инициированным, может пройти через некоторое количество состояний до того, как он завершится.

Разные экземпляры одного и того же класса могут оказаться весьма отличными друг от друга. Шаги, определенные в описании класса, не должны исполняться в предписанном последовательном порядке; они могут выполняться даже параллельно, когда пример конкретизирован.

Можно также сказать, что прецедент может следовать различными путями. Например, экземпляр прецедента Обслуживаниеобеда или Ужина будет действовать особым образом, если у посетителя нет денег заплатить; возможно, у посетителя потребуют документы или даже вызовут полицию. Если при выполнении другого экземпляра того же класса прецедентов у посетителя есть деньги, то спрашивать документы у него не будут. Однако оба эти экземпляра одного прецедента следуют одному описанию, т. е. подчиняются одному классу.

5.2.6

Взаимодействие между субъектом и прецедентом

Выше мы упоминали, что субъекты взаимодействуют с системой, посылая стимулы. Чтобы лучше понять субъект, необходимо знать в каком прецеденте он участвует. В П-модели это указывается стрелками между субъектом и прецедентом. Например, чтобы показать,

что субъект Посетитель общается с прецедентом Обслуживание Завтрака, и наоборот, надо определить отношение как Посетителя с этим прецедентом, так и прецедента с Посетителем. Однако не всегда необходимо определять это двумя стрелками. Можно использовать двунаправленные стрелки, что облегчает понимание схемы (см. рис. 5.2).

Итак, завершая рассмотрение П-модели, подчеркнем еще раз, что цель П-модели - внешнее представление системы. Это означает, что модель предназначена в первую очередь для заказчиков и пользователей системы, а не для тех, кто ее реализует. Прецеденты должны выражать, что бизнес должен делать, в то время как внутренняя модель описывает, как бизнес работает, т.е. П-модель - есть *“что модель”*, а внутренняя модель - *“как модель”*.

Учитывая вышеизложенное, перечислим, что надо и чего не надо показывать в П-модели:

- ❶ В модели не показывается коммуникация между потоками событий. Таким образом, экземпляры прецедента не могут общаться друг с другом, иначе пришлось бы описывать интерфейс между ними.
- ❷ Экземпляры одного или различных классов прецедентов, очевидно, будут влиять в бизнесе друг на друга; П-модель не содержит средств показать это. Конечно, такие отношения очень важны, но они показывают детали внутренней работы бизнеса, не интересные людям, для которых предназначена П-модель. Следовательно, эти влияния будут показаны во внутренней модели.
- ❸ В П-модели не показывается параллельность хода событий. Предполагается, что прецедент интерпретируется как один экземпляр и одна транзакция в каждый промежуток времени. Следовательно, транзакции прецедентов описываются так, как если бы они были неделимыми и последовательными.
- ❹ В соответствии со всем сказанным выше, в П-модели показываются только отношения классов.

5.3

Внутренняя объектная модель

П-модель - хорошее средство как для описания требований к бизнесу, так и для представления наглядной картины того, что бизнес выполняет. П-модель иллюстрирует функции бизнеса, его окружение и бизнес-процессы, которые он предлагает внешнему миру. Тем не менее П-модель не дает ясной картины о внутреннем устройстве бизнеса, т.е. о том, как различные виды деятельности реализуют бизнес-процессы, как эти виды деятельности связываются вместе в цепочки процессов, какой вид ресурсов должен использоваться для реализации того или иного вида деятельности и т. д. Для более полного понимания бизнеса по крайней мере командой по реинжинирингу требуется описание его более полное чем то, которое дается П-моделью.

Есть и другие аспекты бизнеса, которые не описываются П-моделью. Очевидно, что бизнес не должен заниматься ничем, что нельзя соотнести с бизнес-процессом; неверно думать, что все, что н а д о выразить, м о ж н о выразить в терминах бизнес-процессов. Специалисты наблюдали много попыток бизнес-инжиниринга, когда пытались исказить понятие процесса так, чтобы этим словом можно было обозначить что угодно. Очевидно, что в расчет следует принимать не только требования, регулирующие бизнес-процессы, но и другие; например, качество продукции (а не качества самого процесса), структура управления компанией, информация, сообщаемая сотрудникам компании. Можно называть все эти вещи процессами, но это не будет иметь особого смысла. Подобные аспекты будем описывать во внутренней модели.

Используются два вида *внутренних* моделей: идеальные или реальные. *Идеальная модель* - это модель, не учитывающая, как компания должна реализоваться на практике. Такая модель не учитывает, например, что компания географически распределена на несколько филиалов. *Реальная модель* учитывает все эти факторы. Она учитывает, что в настоящее время компания не располагает персоналом, имеющим уровень компетенции, предполагаемый идеальной моделью. Во многих случаях достаточно построить идеальную модель и предоставить самой компании возможность решать, как ей следует работать, чтобы приблизиться к реальной модели (см. гл. 7).

В соответствии с современными тенденциями при описании внутренней модели будем базироваться на понятии "объект", поэтому будем называть внутреннюю модель - *О-моделью* (*объект-моделью*). Заметим, что и П-модель (если позволяет инструментальное средство) естественно описывать, базируясь на объектном подходе. Использование объектов дает компании прочную архитектуру, которая понятна, поддается изменениям, адаптируема и может использоваться повторно (см. разд. 3.4 и 3.5).

5.3.1

Объекты и их типы

При описании О-модели объекты представляют участников процессов и различного рода сущности (продукция, предметы, задачи), используемые в ходе выполнения процессов. Имя, которое дается классу объектов, должно как можно более ясно выражать суть, свойственную экземплярам этого класса. Классам часто дают имена, состоящие из нескольких слов, так как ясность важнее, чем краткость. Имя модели должно быть уникальным, не следует давать классам - или чему-либо еще - похожие имена. Имена классов, как правило, выражаются существительными.

Пример. При описании внутренней О-модели для бизнес-системы Ресторан уместно ввести по крайней мере следующих участников: Гардеробщик, Официант и Повар. (На самом деле этот перечень не полон. В зависимости от моделируемых задач могут потребоваться такие участники, как бухгалтер, уборщица и т.п.). К сущностям, используемым в ходе бизнеса, следует отнести Меню и Заказанные блюда (для краткости - Заказ). Итак, для описания бизнес-системы Ресторан вводим следующие классы объектов: Гардеробщик, Официант, Повар, Меню и Заказ (рис. 5.3).

Объект может соответствовать задачам, продукции или сущностям в бизнесе. Задачи могут быть подразделены на два типа: те, что обеспечивают взаимодействие субъектов с бизнесом, и те, которые являются чисто внутренними. Удобно определить различные типы объектов, для того чтобы сделать более ясными задачи, которые они выполняют в модели. Обычно различают следующие типы объектов: объекты-сущности, управляющие объекты и интерфейсные объекты. Все классы объектов будем изображать в виде треугольников с добавлением букв внутри треугольника (см. рис.5.3): *и* - интерфейсные, *у* - управляющие.

Интерфейсные и управляющие объекты представляют задачи, а не типы ресурсов. С другой стороны, эти задачи выполняются людьми, относящимися к определенной категории ресурсов. В ресторане задача по приготовлению пищи выполняется человеком, обученным готовить (поваром). Часто, однако, несколько задач помещаются вместе в один объект, который реализуется одним конкретным экземпляром ресурса. Например, официант может отвечать как за размещение посетителей за столиками, так и за обслуживание посетителей.

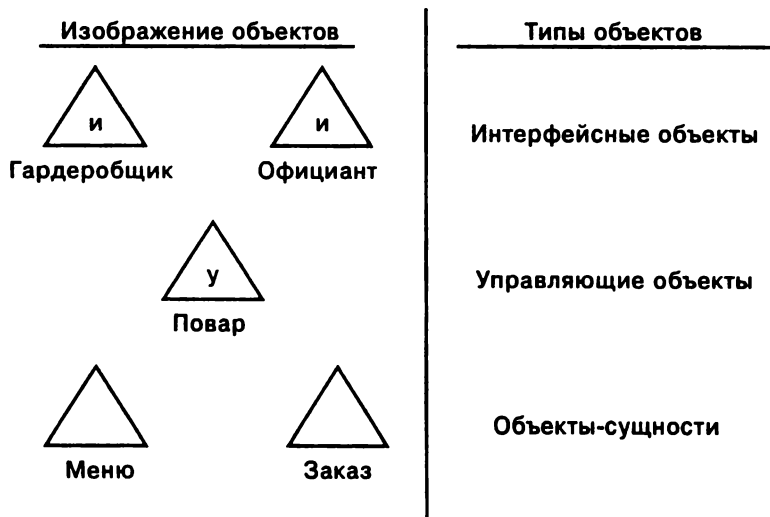


Рис.5.3. Классы и типы в бизнес-системе Ресторан

Интерфейсные объекты представляют в бизнесе операции, каждая из которых должна выполняться одним и тем же ресурсом. Эта задача включает взаимодействие с окружением бизнеса. Следовательно, к коммуникабельности людей, выполняющих эту задачу, предъявляются определенные требования.

Интерфейсный объект может участвовать в нескольких прецедентах. Часто объекты этого типа несут ответственность за координацию процесса в той его части, которая касается непосредственного контакта с клиентами. Каждый конкретный клиент должен иметь как можно меньше контактов с компанией. Это, впрочем, не противоречит бизнес-теориям, которые рекомендуют работникам компании быть ближе к клиенту, а просто означает, что в интересах клиента его контакты с компанией должны быть как можно проще и реже.

Можно сказать, что часть бизнеса, имеющая непосредственный контакт с внешним миром, видима как интерфейсные объекты, в то время как объекты-сущности и управляющие объекты более независимы от окружения.

Управляющие объекты, как и интерфейсные объекты, представляют операции в бизнесе. Различие заключается в том, что эти операции не имеют непосредственного контакта с окружением бизнеса. Название “управляющий объект” используется потому, что эти объекты активны, они управляют или принимают участие в потоке управления при обработке продукции. Следовательно, экземпляр управляющего объекта часто имеет то же время жизни, что и экземпляр прецедента, в котором он участвует. Типичные примеры управляющих объектов в компании - это Разработчик Продукции и Менеджер Проекта.

Объекты-сущности представляют такие сущности, как продукция и предметы, которые обрабатываются бизнесом. Объект-сущность участвует не только в одном прецеденте; экземпляр может принимать участие во многих событиях в бизнесе. В отличие от интерфейсных и управляющих объектов, объекты-сущности представляют сущности, не являющиеся человеческими или техническими ресурсами. Типичные примеры объектов-сущностей в компании - Продукция, Извещение (Invoice) и Заказ.

5.3.2

Отношения

Отношения, существующие в прецедентах, устанавливаются между объектами (классами) и изображаются стрелками (дугами). Отношение связывает два объекта (экземпляра или класса). *Отношение*

ссылки обычно строятся от одного экземпляра к другому, следовательно, говорят, что это отношение экземпляров. *Отношение наследования* связывает два класса, и поэтому его называют отношением классов. Также может потребоваться соотнести класс с экземпляром, или наоборот; этот вид связи называют *мета-отношением*.

Как правило, используют бинарные направленные отношения. Если необходимо показать, что два объекта могут взаимодействовать в обоих направлениях, то используют двунаправленную стрелку (дугу).

Отношение ссылки между объектами. При обслуживании клиентов естественно иметь возможность определить для каждого конкретного заказа, какие счета ему соответствуют. Следовательно, объект Заказ должен содержать одну или более ссылок на экземпляры объекта Счет. Ссылка такого вида между объектами обозначается отношением ссылки. Отношение ссылки между объектом **А** и объектом **В** определяет тип отношения между **А** и **В**. С отношением можно связать количество экземпляров, которые могут быть с ним ассоциированы. В примере, представленном на рис. 5.4, отношение ссылки называется *“способ оплаты”* и указано, что по заказу может быть выписано как 0 счетов, если еда (заказ) доставляется на дом, так и несколько, если посетители хотят по одному заказу платить отдельно.

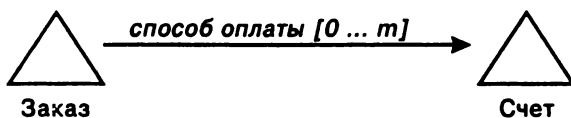


Рис.5.4. Экземпляр класса Заказ ссылается на экземпляр класса Счета

Обычный способ именования отношений таков, что цепочка “объект - отношение - объект” может иметь словесное выражение. Так как названия классов обычно существительные, то имена отношений при этом способе будут глаголами. Например, из последовательности Заказ - Готовит - Повар можно построить фразу “Заказ готовится поваром”.

Однако в [5] предлагается использовать другой способ именования отношений - имя отношения, связывающего **А** с **В**, где отношение играет роль объекта **В** по отношению к объекту **А**. Таким образом, имя отношения выражается примерно так, как показано на рис. 5.5: “Заказ - ответственный за приготовление - Повар”, что читается следующим образом: “Повар отвечает за приготовление заказа”.

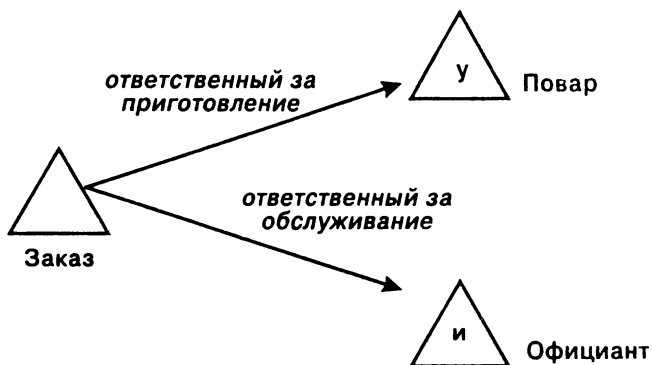


Рис.5.5. Именование отношений

Данный способ именования отношения объясняется двумя причинами.

❶ В объектно-ориентированном мире все рассматривается с точки зрения объекта. В этом мире естественно спросить: “Как один объект зависит от других объектов?” Ответ на этот вопрос может дать понятие о ролях, которые другие объекты играют по отношению к данному. Следовательно, вполне логично давать отношениям имена, отражающие их роли. Также можно сказать, что объект А предъявляет требования к ролям, которые другие объекты должны играть по отношению к А. Однако этим объектам вовсе не обязательно знать, кто или какие предъявляет к ним требования. Таким образом, отношения обычно действуют в одном направлении.

❷ Введение имени отношения, определяющего роли других объектов, способствует структурированию модели. Например, если при начальном описании с объектом Заказ был связан один объект - Работник Ресторана, а затем выяснилось, что с объектом Заказ связано два отношения: “обслуживать” и “готовить”, то ясно, что требуется два разных экземпляра Работник Ресторана. Чтобы не путать классы с отношениями, имена классов будем начинать с заглавной буквы, а отношений - со строчной.

Агрегаты объектов. Отношения, называемые “состоит из” и “является частью”, представляют собой варианты отношения ссылки. Они используются для выражения того, что объект состоит из других объектов. Конструкция такого типа называется *агрегатом*. Отношение “состоит из” связывает объект А с объектом Б и указывает, что А состоит из объектов типа Б (рис. 5.6). Отношение “является частью” выражает обратную ситуацию, т. е. объекты типа Б являются частью объектов типа А. Как правило, отношения “состоит из”, “является частью” существуют между экземплярами.

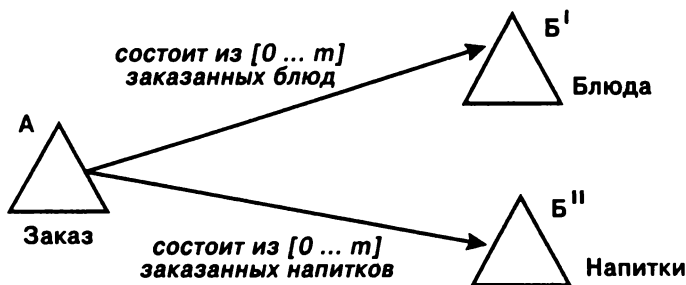


Рис.5.6. Заказ состоит из некоторого количества блюд и напитков

Отношение коммуникации. Объекты должны иметь возможность обмениваться данными. В нашем примере объект Повар должен узнать из объекта Заказ, какие блюда он должен готовить. Этот тип отношения выражается отношением коммуникации между двумя объектами. Так же, как и для отношения ссылка, для отношения коммуникации указывается количество экземпляров, которые можно с ним ассоциировать. Объект Повар может готовить от 0 до $m > 0$ экземпляров Заказа, что изображается в виде диапазона $[0...m]$, приписываемого отношению (рис. 5.7).

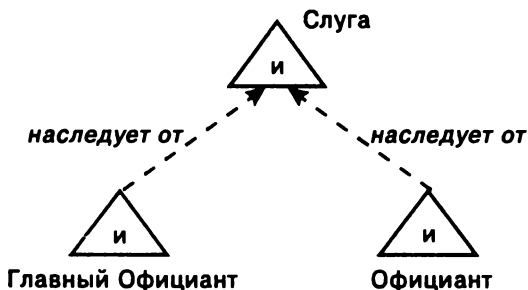


Рис.5.7. Классы Главный Официант и Официант наследуют характеристики от класса Слуга

Отношение коммуникации между двумя объектами означает, что они могут “разговаривать” друг с другом. Направление отношения показывает направление передачи информации (стимулов). Отношение коммуникации почти всегда является отношением между экземплярами.

Отношение коммуникации не показывает, какой вид информации передается между объектами; этот уровень детализации может быть представлен с помощью диаграмм взаимодействия (см. п. 5.4.2). Отношение лишь показывает, что коммуникация имеет место. Именно поэтому этим отношениям не дают имен.

Отношение наследования. В ресторане метрдотель (Главный Официант) отвечает за прием посетителей, распределение мест (столиков) и за другие общие вопросы. Метрдотель и Официант имеют общие характеристики. Чтобы не описывать эти свойства несколько раз, общие части этих классов помещают в новый класс, называемый “Слуга”, от которого классы Официант и Главный Официант наследуют характеристики (рис. 5.7). Можно сказать по-другому: классы Главный Официант и Официант являются подклассами более общего класса Слуга.

Отношение наследования - это отношение между классами. При конкретизации наследуемого класса (потомка) его экземпляр получает все характеристики (атрибуты, операции и отношения), описанные в его суперклассе.

5.3.3

Поведение

Изучая прецеденты, в которых участвует объект, можно получить представление об обязательствах объекта по отношению к его окружению. Поведение, описываемое в классе объекта, должно подтверждать эти обязательства. Поведение объекта можно разделить на несколько последовательностей, называемых *операциями*. Выполнение каждой такой операции инициируется определенным стимулом. Чтобы выполнить операцию, часто бывает нужно знать определенные входные и выходные данные, которые называют *параметрами операции*. Стимул вместе с параметрами образует *сигнатуру операции*.

Набор операций для класса объектов - *протокол класса* - показывает, как экземпляры других классов могут посылать стимулы экземплярам данного класса. Структура класса, так же, как и описание его поведения, инкапсулирована в самом классе (см. разд. 3.5).

Не всегда требуется выражать поведение объекта в терминах операций и параметров. Если основная задача объектной модели - идентифицировать задачи, продукты и предметы, не описывая их подробно, то скорее всего будет достаточно дать короткое неформальное описание поведения. Важно, чтобы обязательства объекта по отношению к его окружению были ясно указаны.

Например, объект Заказ должен (по крайней мере) быть способен указать Блюдо, которое следует приготовить, и Официанта, который за него отвечает. У Заказа можно идентифицировать две операции: Какое Блюдо и Какой Официант. Стимул, вызывающий операцию Какое Блюдо, имеет параметр ответа - название блюда и т.п.

5.3.4

Атрибуты

Характеристики объекта моделируются атрибутами объекта. *Атрибут* представляет собой единицу информации, хранящуюся в объекте, и состоит из отношения “быть атрибутом” и его типа. Отношение “атрибут” представляет значение атрибута, в то время как тип атрибута обозначает его структуру и тип. Отношение “атрибут” имеет имя, описывающее роль, которую атрибут играет по отношению к объекту. Отношение также может иметь мощность, указывающую, сколько экземпляров атрибута может быть ассоциировано с этим отношением. Отношение “атрибут” обычно связывает экземпляр класса объектов с экземпляром типа атрибута.

В объектной модели можно описывать не все атрибуты объекта, а только те, которые нужны для понимания роли объекта и его обязательств в бизнесе. Например, объект Заказ может иметь атрибуты, указывающие, какие блюда и напитки заказаны. Если понятия Блюдо и Напиток нужны только для заказа, то этот вариант предпочтительнее, чем вводить отношение Заказ “состоит из” Блюда и Напитков.

5.3.5

Состояния объектов

Объект может получать различные стимулы в зависимости от значений атрибутов и от предварительно выполненных операций, т.е. объект имеет различные состояния. Рассмотрим объект Повар из нашего примера. Если Повар имеет много заказов, например десять (рис. 5.8), будем считать, что он находится в состоянии “полностью занят”, а в остальных случаях Повар находится в состоянии “не полностью занят”. То, в какое состояние совершит переход объект, зависит от его текущего состояния, и от того, какой стимул он получит.

Легко представить, что объект может иметь гораздо более сложные состояния, чем те, что описаны в примере. Следовательно, целесообразно изобразить диаграмму состояний и переходов. Для построения таких диаграмм разработано много методик; выбор методики зависит в первую очередь от того, насколько сложны состояния. Диаграмма состояний и переходов соотносит события с состояниями. Когда происходит событие, следующее состояние объекта зависит частично от его настоящего состояния, частично от происшедшего события. Простой способ построения диаграмм состояний и переходов представлен на рис. 5.8; этот тип диаграмм обычно называют *диаграммами Мили* [9].

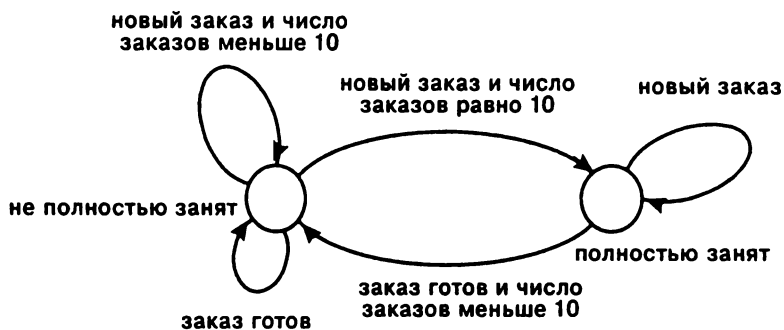


Рис.5.8. Диаграмма состояний и переходов для объекта Повар

5.4

Прецеденты и объекты

П-модель и О-модель - это разные типы описания бизнеса. О-модель, состоящая из сотен объектов и более, неудобна для работы и малопонятна, так как слишком сложна. Она содержит детали того, как объекты связаны друг с другом, как они общаются, какой интерфейс используют, что они знают друг о друге и т. д. Информация такого рода делает любое описание слишком детальным, и есть риск, что больше энергии будет тратиться на объяснение внутреннего взаимодействия, чем на понимание бизнес-процессов. Конечно, информация такого рода очень важна при создании подробных моделей бизнес-систем, однако она не обязательна для понимания прецедентов.

Как указывают многие специалисты [7], [8], выявлять объекты для модели не сложно, сложно н а х о д и т ь правильные объекты. Рекомендуется для каждого прецедента определить объекты, необходимые для его выполнения. Если в объектную модель включать объекты, принимающие участие в различных прецедентах, то будет больше шансов определить действительно нужные объекты. Рассмотрев все прецеденты, в которых некоторый объект играет какие-либо роли, можно понять назначение этого объекта в системе [10].

П р и м е р . Прецеденты Обслуживание Завтрака, Обслуживание Обед а или Ужина, Обслуживание Мероприятия, Поставка Продуктов используют набор объектов. Некоторые объекты принимают участие в нескольких прецедентах: например, Повар участвует во всех. Назначение объекта Повар проявляется при рассмотрении всех прецедентов, в которых он используется.

Если необходимо более подробно показать, как объекты взаимодействуют при выполнении прецедентов, то для этого привлекают, например, диаграммы взаимодействий (см. подразд. 5.4.2).

Более того, именно посредством объектов прецеденты воздействуют друг на друга. Два экземпляра прецедента одного или двух разных классов могут быть реализованы при помощи одного и того же экземпляра объекта. В описанном выше примере управляющий объект Повар участвует во всех четырех прецедентах. Вследствие этого могут возникнуть конфликты. Например, если Повар участвует в двух последовательных действиях при Обслуживании Завтрака, то его не следует использовать в перерыве в прецеденте Поставка Продуктов, если есть какой-либо риск возникновения несогласованности. Следовательно, транзакцию в прецеденте необходимо рассматривать как целое, и не следует ее прерывать или разделять.

Другой вид конфликта может возникнуть при использовании одного экземпляра объекта-сущности двумя различными экземплярами прецедентов. Если, например, некоторый документ представлен как объект-сущность и этот документ участвует в нескольких экземплярах прецедентов, должно быть ясно указано, кто несет за него ответственность, т.е. имеет право изменять состояние экземпляра. Пример с рестораном, однако, не демонстрирует эту проблему. Она возникает, когда объекты-сущности являются носителями информации, например, объект Счет в модели банка и т.п.

5.4.1

Взаимодействие объектов в прецеденте

Взаимодействие объектов в прецеденте (рис. 5.9) отображает динамику объектной модели. Этот тип представления содержит, помимо объектов, отношения коммуникации, необходимые для выполнения прецедента. Чтобы объяснить, кто и как инициирует потоки событий, необходимо включить в представление субъекты и отношения, связывающие субъекты с объектами.

Так как объект участвует в нескольких прецедентах, он будет присутствовать в нескольких представлениях взаимодействия объектов. Вместе эти представления иллюстрируют различные роли, которые объект имеет в системе. Иначе говоря, каждый прецедент, в котором участвует объект, накладывает на него обязательства (ответственность).

Человеку, как известно, свойственно забывать, почему выбрано то или иное решение данной проблемы, поэтому важно документировать выбранное “движение” прецедента между объектами. Следовательно, необходимо создать еще одно описание потока событий данного прецедента в терминах участвующих объектов. Подчеркнем, что речь идет о еще одном описании прецедента, а не о новом преце-

денте. Такое описание прецедента Обслуживание Обеда или Ужина будет выглядеть следующим образом (см. рис. 5.9):

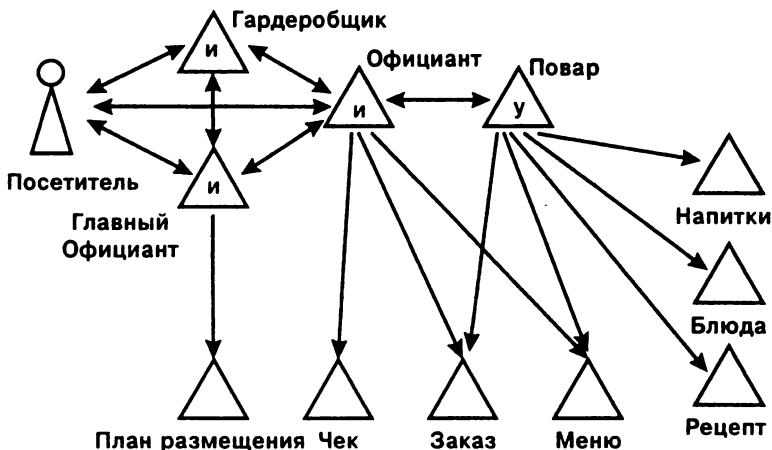


Рис.5.9. Объекты, участвующие в прецеденте Обслуживание Обеда или Ужина

А. Войдя в ресторан, субъект Посетитель посылает стимул, означающий, что он хочет, чтобы его обслужили.

Б. Объект Гардеробщик спрашивает Посетителя, желает ли тот сдать пальто. Как вариант: Посетитель сам может проинформировать Гардеробщика о своем желании. После этого Посетителя встречает Главный Официант, который сажает его за столик в соответствии с Планом Размещения. Затем Главный Официант вызывает Официанта, который будет обслуживать Посетителя. Официант дает Посетителю меню или на словах описывает, что сегодня подают. Предложенные варианты соответствуют текущему экземпляру объекта Меню.

В. После соответствующего промежутка времени объект Официант просит Посетителя сделать заказ. Альтернатива: Посетитель сам может позвать официанта. Посетитель делает заказ и Официант проверяет, все ли из заказанного доступно в Меню.

Г. Официант заносит заказ посетителя в новый экземпляр объекта Заказ. Затем Официант создает новый экземпляр объекта Счет. Официант информирует экземпляр объекта Повар о том, что поступил новый заказ.

Д. Объект Повар готовит различные блюда в соответствии с Рецептами. Он также достает из холодильника заказанные Напитки.

Е. Когда еда готова, Повар информирует об этом Официанта, обслуживающего Посетителя. Когда Посетитель кончает есть, ожидается, что он сообщит это некоторому экземпляру объекта Официант.

Ж. Официант обновляет объект Счет и представляет его Посетителю. Ожидается, что Посетитель заплатит. Затем Посетитель просит объект Гардеробщик вернуть оставленные на хранение вещи и покидает ресторан. На этом прецедент завершается.

Альтернативы хода событий:

• **А л ь т е р н а т и в а А**: если на этапе **А** ресторан полон, объект Гардеробщик сообщает об этом субъекту Посетитель. Посетителю говорят, что он может посидеть в баре, пока освободится столик. Когда объект Главный Официант сообщает объекту Гардеробщик, что освободился столик, он говорит об этом Посетителю. В этом случае прецедент продолжается с этапа **Б**. Если Посетитель не хочет ждать и уходит, то прецедент завершается.

• **А л ь т е р н а т и в а В**: если на этапе **В** блюдо, которое заказал Посетитель, не включено в объект Меню, то объект Официант говорит Посетителю, что это блюдо сегодня не подается. Независимо от того, по какой причине это блюдо недоступно: из-за несоответствий в напечатанном меню или потому, что у Посетителя какие-то особые требования, Официант должен уметь предложить замену. Когда субъект Посетитель и объект Официант приходят к соглашению, прецедент продолжается с этапа **Г**.

Заметим, что мы следовали той же структуре, что и в первом описании прецедента (см. подразд. 5.2.5). Так легче отслеживать объекты и подпотоки событий в прецеденте.

5.4.2

Диаграммы взаимодействий

Если необходимо более подробно описать, как при выполнении потока событий взаимодействуют объекты модели, можно построить диаграмму взаимодействий (см., например, [5]). Эта диаграмма показывает, как взаимодействующие объекты реализуют прецедент. При этом идентифицируются стимулы, передаваемые между объектами, и параметры этих стимулов, т.е. идентифицируются протоколы взаимодействия объектов.

В диаграмме взаимодействий объект представляется вертикальным столбцом (рис. 5.10). Порядок расположения столбцов неважен, их просто следует размещать так, чтобы все выглядело нагляднее. Если в прецеденте присутствует несколько экземпляров одного класса, каждый из них можно представить отдельным столбцом. Столбец чаще всего представляет экземпляр, но может представлять и класс.

Ось времени в диаграмме взаимодействий идет сверху вниз. Ее не следует понимать как линейную, она определяется стимулами. Другими словами, расстояние между двумя стимулами на диаграмме не имеет ничего общего с интервалом времени между событиями.

С левой стороны диаграммы взаимодействий обычно описывают поведение, выполняемое объектами (пути операций). В столбцах пути операций представляются жирными вертикальными отрезками. Описания путей операций дают основу для идентификации и описания операций объектов. Не всегда практично делать описания всех путей операций, так как структура текста на полях может стать слишком сложной. В нашем примере описаны пути операций только для интерфейсных и управляющих объектов, так как именно в этих объектах обрабатываются стимулы.

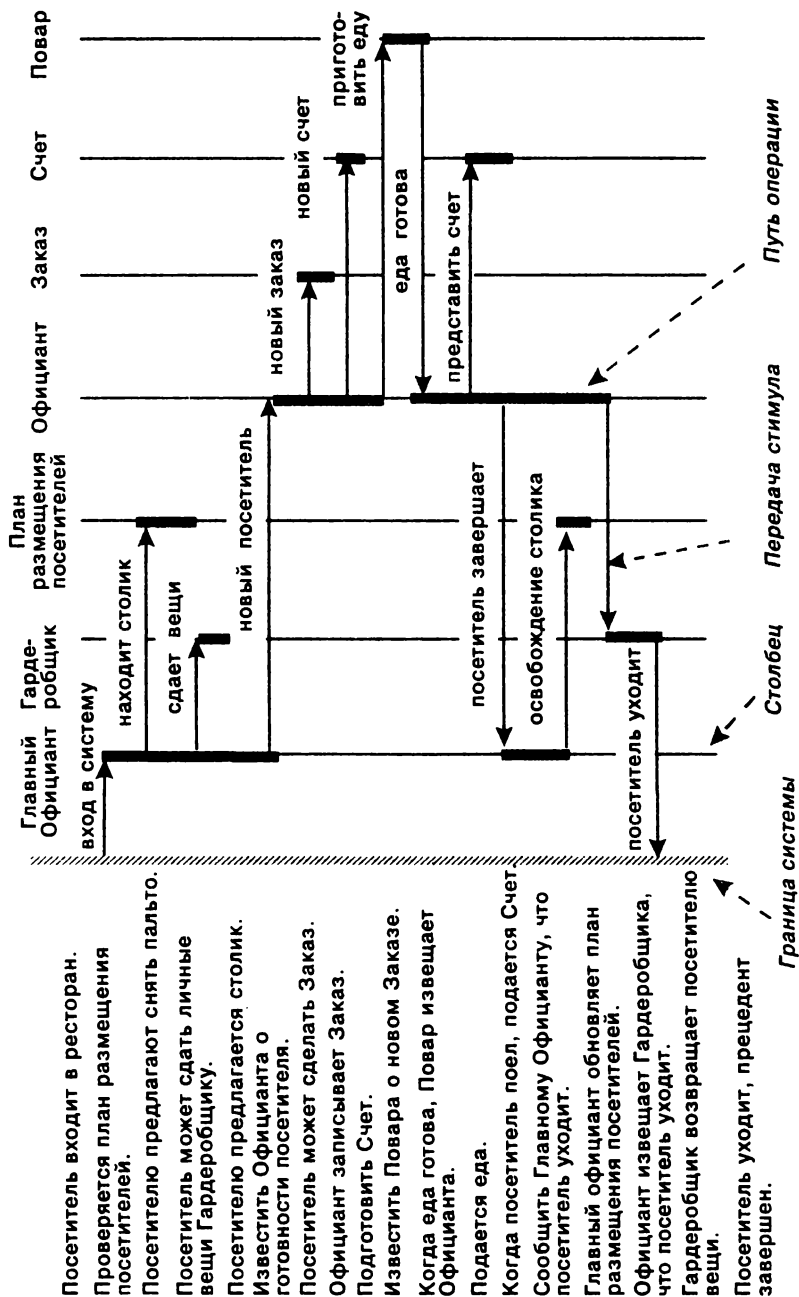


Рис. 5.10. Диаграмма взаимодействий для прецедента Обслуживание Обеда или Ужина

Прецедент обычно содержит несколько вариантов хода событий. Не стоит пытаться изобразить их все на одной диаграмме взаимодействий, так как она станет слишком запутанной. Лучше построить для прецедента несколько диаграмм, где каждая будет наглядно показывать, что происходит при альтернативном ходе событий или во фрагменте потока событий.

Методика построения диаграмм взаимодействий использовалась долгое время в сфере телекоммуникаций, в основном, для описания взаимодействия между блоками аппаратуры. И.Якобсон в 1987 г. предложил использовать эти диаграммы для объектно-ориентированных методов [3].

5.5

Отношения между прецедентами

Когда система становится большой, важно создать ее исчерпывающую модель. Целесообразно сначала показать основные прецеденты модели, а затем добавить вторичные потоки событий. Их можно выразить как в виде новых прецедентов, так и в виде расширения основных прецедентов. Чтобы показать, что один прецедент может расширять другой, вводят *отношение расширения* [4]. При создании П-модели бизнеса вполне может оказаться так, что два прецедента содержат похожие описания. Чтобы избежать таких наложений, вводят *отношение использования*.

5.5.1

Отношение расширения

Описание прецедента может быть довольно сложным для понимания, если оно содержит слишком много альтернативных, необязательных (условных) или исключительных потоков событий, выполняемых только при определенных условиях. Один из способов сделать описание "прозрачнее" - извлечь некоторые его фрагменты и рассматривать их как отдельный прецедент. Говорят, что этот новый пример расширяет первоначальный, если выполнены требуемые условия. Такую конструкцию можно получить, используя отношения расширения между прецедентами.

Отношение расширения между прецедентом А и прецедентом В (А->В) означает, что экземпляр, соответствующий описанию В, может, если выполнены требуемые условия, соответствовать описанию А. Когда поток событий, детализированный описанием А, завершен, экземпляр снова будет соответствовать описанию В. В описании расширяющего прецедента А должно быть указано, в дополнение к

описанию потока событий, при каких условиях А должен быть вставлен в расширяемый прецедент В.

Пример. В Ресторане посетители могут получить: закуску, десерт или заказать ужин при свечах, когда они сидят в отдельном кабинете с музыкальным сопровождением. Ранее был определен прецедент Обслуживание Обеда или Ужина, который можно расширить, добавив прецеденты Обслуживание Закуски, Обслуживание Десерта и Ужин при Свечах (рис. 5.11).

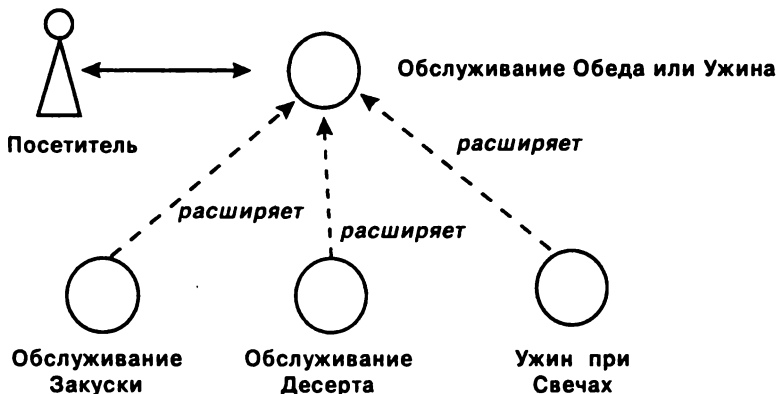


Рис.5.11. Отношение расширения между прецедентом Обслуживание Обеда или Ужина и другими прецедентами означает, что только прецедент Обслуживание Обеда или Ужина будет инсталлироваться независимо

Таким образом, отношение расширения используется для того, чтобы:

- показать условные части прецедента;
- более подробно смоделировать сложный и (или) альтернативный ход событий;
- смоделировать фрагменты потока событий, выполняемые в особых случаях;
- смоделировать добавление в поток событий нескольких различных фрагментов в любой комбинации. Эти фрагменты рассматриваются как самостоятельные прецеденты.

На рис. 5.11 Обслуживание Закуски и Обслуживание Десерта относятся к первому типу расширения, а Ужин при Свечах - к третьему. Другие фрагменты, которые можно моделировать как расширяющие прецеденты - это, например, ситуация, когда у посетителя нет денег заплатить или он требует приготовление блюда по определенному рецепту.

Отношение расширения можно рассматривать как способ структурирования потока событий.

Отношение использования

При описании прецедентов можно обнаружить, что некоторые из них имеют общие фрагменты. Чтобы не описывать эти общие фрагменты более одного раза, надо их описать в виде самостоятельного прецедента, который затем может использоваться остальными. Это обозначается в модели отношением использования. Отношение использования помогает избежать лишних описаний, позволяя повторно применять общие фрагменты потока событий.

Отношение использования между одним прецедентом **A** и другим прецедентом **B** ($A \rightarrow B$) показывает, что экземпляр **A** выполняет поток событий, описанный для **B**. Отношение использования применяется, когда нужно показать, что описание транзакции совместно используется несколькими прецедентами.

Пример. В прецедентах Обслуживаниеобедаилиужина и Обслуживаниезавтрака выполняются процедуры ОбслуживаниезакускииДесерта. Так как эти процедуры идентичны для обеда, ужина и завтрака, то этот поток событий можно выделить в отдельный прецедент, который используется как при Обслуживанииобедаилиужина (рис. 5.12), так и при ОбслуживанииЗавтрака.

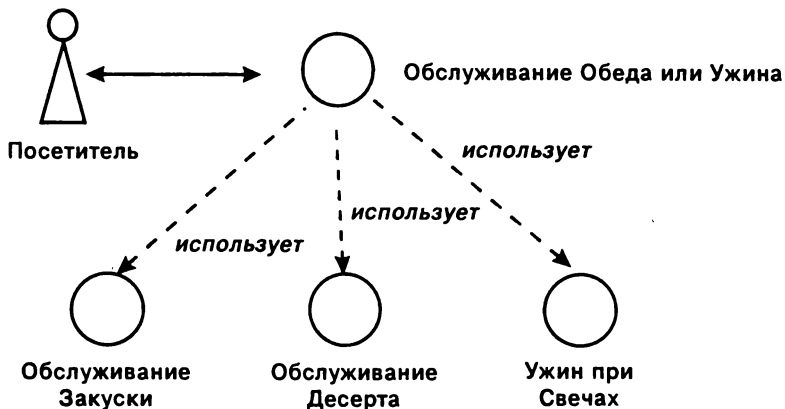


Рис.5.12. Отношение использования при Обслуживанииобедаилиужина

Отношение использования подобно отношению наследования: оба представляют собой повторное применение описания. Прецедент может иметь отношения использования с несколькими другими прецедентами - сравните это с множественным наследованием.

Различие отношений использования и расширения

Отношение использования и отношение расширения можно рассматривать как вид наследования. Рассмотрим разницу между ними на следующем примере.

Пример. Обслуживание Закуски, Обслуживание Десерта и Ужин при Свечах имеют отношение расширения к Обслуживаниюобеда или Ужина. Когда устанавливается Обслуживаниеобеда или Ужина, то его экземпляр может на определенном этапе жизненного цикла следовать описанию следующих прецедентов: Обслуживание Закуски, Обслуживание Десерта или Ужин при Свечах. Однако это происходит только при определенных условиях, т. е. когда посетитель просит одно или несколько из этих расширений. Таким образом, Обслуживание Закуски, Обслуживание Десерта и Ужин при Свечах сами по себе в этой модели никогда не вызываются (см. рис. 5.11).

Что произойдет, если вместо отношения расширения выбрать отношение использования? Выделим два аспекта:

① Обслуживаниеобеда или Ужина связано отношением использования с тремя прецедентами (см. рис. 5.12). В этом случае (в отличие от применения отношения расширения) понимание прецедента Обслуживаниеобеда или Ужина невозможно вне зависимости от прецедентов Обслуживание Закуски, Обслуживание Десерта и Ужин при Свечах. Преимущество отношения расширения в том, что оно позволяет делать различные изменения в ходе событий основного прецедента, не принимая во внимание, как это отразится на расширенных потоках событий.

② Обслуживание Закуски, Обслуживание Десерта и Ужин при Свечах используются в прецеденте Обслуживаниеобеда или Ужина (см. рис. 5.12). В данном случае модель не будет эквивалентна той, что изображена на рис. 5.11, так как в этой модели (представлена на рис. 5.12) ресторан сможет предлагать только некоторые из вариантов обслуживания (см. рис. 5.11) - основной набор с закуской, основной набор с десертом или ужин при свечах только с основным набором. Чтобы получить все те функции, что представлены на рис. 5.11, придется определить прецеденты для всех комбинаций основного набора, закуски, десерта и ужина при свечах.

В большинстве случаев выбрать между отношениями использования и расширением несложно. Отношение использования применяется для выделения общих последовательностей в нескольких прецедентах, в то время как отношение расширения применяется для добавления новых фрагментов хода событий.

Подсистемы и прецеденты

Перед тем как начать подробно описывать ход событий бизнеса, целесообразно разделить его на меньшие компоненты. В большом бизнесе можно выделить много частей (подбизнесов), которые требуют различной разработки, выполнения и сопровождения. Выделение частей бизнеса называют *определением высокоуровневой архитектуры бизнеса*. Для описания архитектуры используют понятие “подсистема”. Если бизнес очень большой, имеет смысл определить несколько уровней подбизнеса (функций).

Подсистема включает функционально близкие объекты и (или) подсистемы. При этом объект или подсистема не может принадлежать более чем к одной подсистеме. Деление на подсистемы следует проводить тогда, когда бизнес настолько велик, что его необходимо моделировать не весь сразу, а постепенно. Кроме того, выделение подсистем целесообразно использовать, чтобы сделать модель понятнее. Людям, участвующим в конкретной части бизнеса, легче будет понять общую картину, если модель покажет, как выполняемая ими часть бизнеса связана с другими частями.

Подсистемы можно использовать для объединения объектов в модули, соответствующие полномочиям некоторого специалиста, отвечающего за одну или более задач.

Пример. Ресторан состоит из *Кухни*, где руководит шеф-повар, и *Столовой*, за которую отвечает главный официант. Важно уяснить, как персонал в кухне и в столовой общается друг с другом и как распределяются их обязанности. Чтобы показать это, выделим две подсистемы - Столовая и Кухня (рис. 5.13).

Можно показать, как эти подсистемы зависят друг от друга при помощи отношения “*зависит от*”. Если объект в подсистеме А имеет отношение к объекту в подсистеме В, то подсистема А “зависит от” подсистемы В.

В чем разница между прецедентами и подсистемами? Оба понятия можно рассматривать как группирование объектов, однако прецедент представляет главным образом описание потоков событий, которые должна реализовывать бизнес-система. Объект может относиться к нескольким прецедентам, но только к одной подсистеме. В подсистеме объекты собраны в соответствии с их функциями. Прецедент, наоборот, может выполняться объектами из разных подсистем. Так что можно сказать, что прецедент пересекает несколько подсистем.

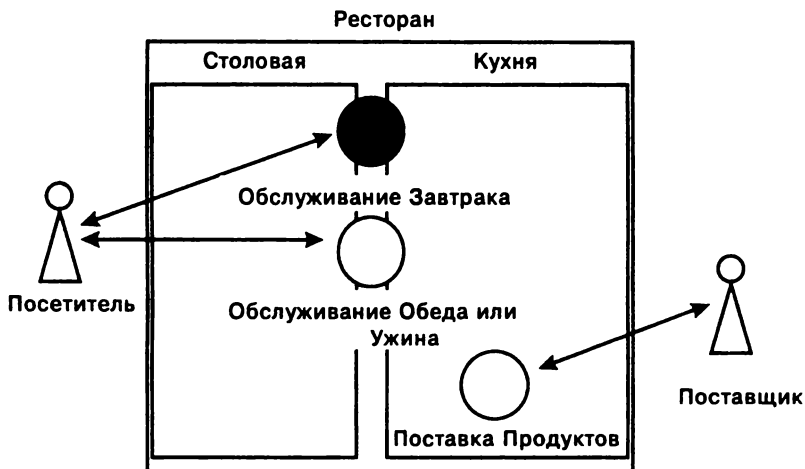


Рис.5.13. Две подсистемы Ресторан: Столовая, Кухня.
Прецеденты могут пересекать несколько подсистем

5.7

Масштабирование моделей бизнеса

Очевидно, что большую компанию типа IBM, Xerox, Ericsson или АBB гораздо сложнее моделировать, чем маленькую компанию. Разница между большим и малым бизнесом состоит в количестве событий и объектов, из-за чего можно легко погрузиться в детали и потерять общую картину. Необходимо уметь расширять (масштабировать) используемый метод моделирования так, чтобы можно было иметь дело со сложным бизнесом - бизнесом, который распределен географически (как национально, так и интернационально) и инфраструктура которого сложна и специализирована. В этом разделе кратко описываются несколько методик, поддерживающих переход к большему масштабу. Эти методики могут применяться независимо одна от другой и в различных комбинациях.

- Начнем с описания того, как можно использовать несколько П-моделей бизнес-системы для различных участников и на различных уровнях абстракции.

- Далее покажем, как можно описывать компании, имеющие несколько независимых сфер деятельности, которые вместе обслуживают общего клиента.

- Наконец, опишем, как использование многоуровневых моделей помогает выделить общие (повторно используемые) и специфические компоненты бизнеса для компании.

Опыт в применении подобных методик довольно ограничен. Приводимые ниже схематические примеры служат для того, чтобы пояснить общие идеи. Работы в этом направлении будут безусловно развиваться и детализироваться специалистами ведущих компаний (см., например, [5]).

5.7.1

Несколько П-моделей на различных уровнях абстракции

П-модель - удобный инструмент для моделирования бизнес-процессов компании. В П-модели заинтересованы как исполнительное руководство компании, так и владельцы процессов. Руководству П-модель нужна, чтобы работать со стратегическими целями компании, а владельцам процессов и лидерам процессов - чтобы иметь детальную картину того, как должны выполняться конкретные процессы. Руководству требуется более общая картина, чем владельцам процессов и лидерам процессов. Этого можно добиться созданием П-моделей на разных уровнях абстракции.

Уровень абстракции и количество деталей в модели определяются выбором субъектов и прецедентов. Например, в П-модели для исполнительного руководства субъект Рынок может соответствовать прецеденту Маркетинг, но этот уровень будет слишком абстрактным для владельца процесса Маркетинг. Ему потребуется выделить различные типы субъектов, так как Маркетинг выполняется совершенно разными способами в зависимости от продукта, географической области рынка, ожидаемых характеристик покупателя и т.д.

Если различие между требованиями к П-модели исполнительного руководства и владельцев процессов очень велика, то можно разработать разные, но связанные модели. Например, можно начать с разработки общей П-модели, которая позднее трансформируется в более подробную П-модель. При малом размере, вероятно, достаточно оставить только подробную П-модель. Для крупной компании необходимо поддерживать простую (общую) модель наряду с более подробной.

Различные П-модели “предназначены для” различных участников. Вместо термина “предназначен для” используют термин “отвечать за” (own). Тот факт, что некоторая категория участников отвечает за модель, означает, что эта категория имеет над ней контроль, т.е. эти люди определяют, как модель должна выглядеть, они могут ее модифицировать и использовать как инструмент взаимного общения. Это не означает, что другие категории участников не могут использовать эту модель другим способом, например для создания индивидуальной модели с другими целями. Часто создаются не-

сколько моделей компании или информационной системы. Между такими моделями необходимо установить взаимные связи так, чтобы сущности в одной модели могли быть сопоставлены с сущностями в другой.

Методика использования нескольких П-моделей на разных уровнях абстракции также применима, если необходимо проводить реинжиниринг процессов последовательно. Сначала создается общая П-модель всего бизнеса, затем определяются приоритетные прецеденты и для них строятся подробные П-модели.

5.7.2

Главные и подчиненные П-модели

Методика моделирования процессов, представленная в разд. 5.1 - 5.6 данной главы, может быть использована для небольших компаний, деятельность которых сосредоточена в одном месте. Для более крупных компаний, которые имеют несколько сфер бизнеса или распределены географически, эту методику необходимо масштабировать (в сторону увеличения).

Для масштабирования модели бизнеса добавим в П-модель две новые конструкции: сферу бизнеса и фрагмент прецедента (подпрецедент).

Пример. Предположим, что есть П-модель компании А, показанная на рис. 5.14 а.

Компания имеет несколько сфер бизнеса (р, г, s), каждая из которых является независимым источником дохода. Так как они полностью автономны, нет смысла описывать их в одной модели, более подходящий способ - общаться с ними как с отдельными бизнес-системами. Предположим, что эти различные сферы бизнеса вместе обслуживают конкретный тип клиентов, и определим для каждой сферы бизнеса соответствующую ему бизнес-систему. Затем разделим предварительно выделенные прецеденты на фрагменты (подпрецеденты). Каждый фрагмент - это часть главного прецедента, относящаяся к определенной бизнес-системе.

Простая П-модель, показанная на рис. 5.14 а, может быть расширена до модели, изображенной на рис. 5.14 б. Корпорация А разделена на три бизнес-системы: N, M и L, а прецеденты р, г и s - на подпрецеденты: p1, p2, p3; g1, g2 и s1, s2, где pi - это подпрецедент, взятый из прецедента p и относящийся к сфере бизнеса N и т.д.

Будем простую П-модель называть *главной П-моделью*, а расширенную часть (нижняя часть на рис. 5.14 б) - *подчиненной П-моделью*. Отношение между главной и подчиненной моделями интерпретируется как их эквивалентность с точки зрения первоначального клиента. Например, поведение корпорации А по отношению к Клиенту 1 эквивалентно в главной и подчиненной моделях. Это означает, что прецедент р и три прецедента p1, p2 и p3 вместе представляют эквивалентное поведение по отношению к Клиенту 1.

Таким образом, существует своего рода отношение эквивалентности между главной и подчиненной П-моделями. Каждому субъекту в главной П-модели в точности соответствует субъект в подчиненной модели. Каждому прецеденту в главной П-модели соответствует набор подпрецедентов в множестве сфер бизнеса в подчиненной модели. Подпрецеденты связаны вместе при помощи отношения коммуникации. Некоторый подпрецедент не обязательно должен быть связан со всеми остальными подпрецедентами. Так, в нашем примере **p1** и **p3** не связаны друг с другом, но каждый подпрецедент связан хотя бы с одним другим подпрецедентом. Когда говорят о подпрецедентах, то имеют в виду, что они происходят от одного прецедента в главной П-модели. Следует упомянуть (один) особый случай. Вполне возможно, что прецеденту в главной П-модели будет соответствовать всего один подпрецедент в подчиненной модели, т. е. этот прецедент не делится на фрагменты. Наконец, каждый субъект, связанный с прецедентом в главной модели, обычно связан только с одним подпрецедентом в подчиненной модели (теоретически нет причины не связывать субъект с несколькими подпрецедентами, но это обычно не практикуется).

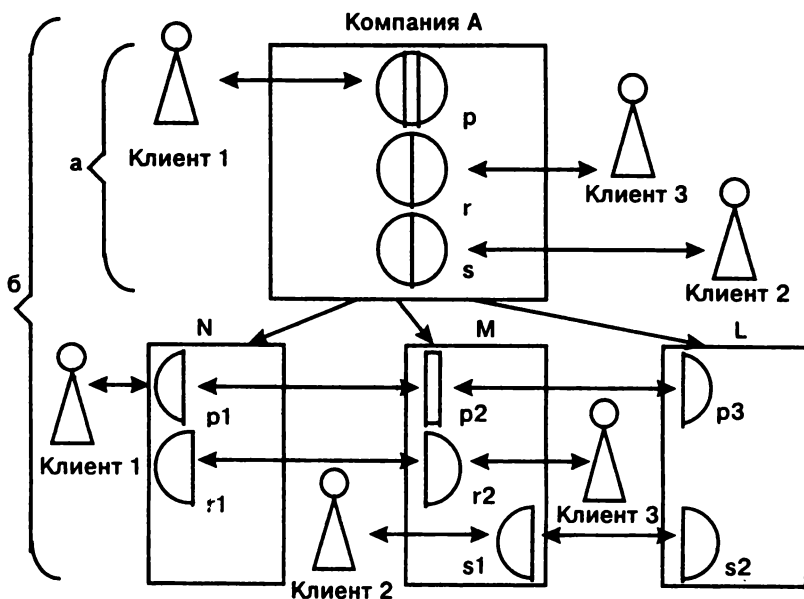


Рис. 5.14. П-модель компании А:
 а - общая П-модель;
 б - П-модель сфер бизнеса

Необходимо описать, как прецеденты в главной П-модели составлены из подпрецедентов в подчиненной модели. Особенно важно определить интерфейс между связанными подпрецедентами.

На этом этапе моделирования можно заняться каждой отдельной сферой бизнеса. Каждая сфера бизнеса рассматривается как бизнес-система, а все ее подпрецеденты - как прецеденты этой системы. Сфера бизнеса N рассматривается как отдельная бизнес-система с прецедентами $p1$ и $r1$. Добавлено два субъекта: Клиент $p2$ Клиент $r2$. С точки зрения N , они ведут себя эквивалентно прецедентам $p2$ и $r2$. Можно сказать, что Клиент $p2$ и Клиент $r2$ изображают $p2$ и $r2$ с точки зрения N .

Итак, бизнес-система N рассматривается как новая система с прецедентами $p1$ и $r1$. Объекты, нужные для реализации прецедентов, определены в точности так, как описано выше в данной главе.

Каково распределение обязанностей в компаниях такого рода? В принципе, владельцы процессов назначаются для всего бизнеса. Так, в нашем примере назначены владельцы процессов p , r и s . Владельцы процессов обеспечивают возможность того, чтобы прецеденты были составлены из подходящих подпрецедентов, обращая особое внимание на интерфейс между подпрецедентами. Затем назначаются владельцы каждого из подпрецедентов. В зависимости от того, как должна выглядеть компания, можно найти различные решения. Если главная модель представляет конгломерат, а сферы бизнеса - филиалы с различными функциями, то владельцы процессов верхнего уровня отвечают за картину в целом и за то, чтобы интерфейс был описан надлежащим образом. Владельцы процессов нижнего уровня отвечают за то, чтобы процессы в филиалах выполнялись надлежащим образом.

Обычно не принято назначать лидеров процессов для прецедентов главного уровня. Однако лидеры процессов назначаются для каждого экземпляра подпрецедентов. Окончательная ответственность за бизнес лежит на руководителях каждой сферы бизнеса. Они не только владеют ресурсами (владельцы ресурсов), но и отвечают за то, чтобы работа в их сфере выполнялась правильно.

5.7.3

Многоуровневые модели бизнеса

Еще одна методика, которую можно использовать для исчерпывающего описания бизнеса, - разбиение на уровни. Эта методика позволяет справиться со сложностью большого бизнеса. Ее достоинства проявляются в основном при моделировании видов деятельности, отличных от бизнес-процессов, таких, как инфраструктурные процессы.

При многоуровневой архитектуре нижние уровни относятся к реализации, т. е. к различным типам ресурсов, в то время как верхние уровни представляют бизнес. Бизнес-система может иметь больше уровней, чем изображено на рис. 5.15, а также может быть разбита на уровни в нескольких измерениях, так что уровень может в свою очередь состоять из нескольких подуровней.

Уровень ресурсов содержит различные конкретные ресурсы, которые нужны для реализации компании. Здесь находятся человеческие ресурсы различных профессий и различные виды оборудования, такого как информационные системы и техника.

Уровень инфраструктуры состоит из объектов, характерных для моделируемого бизнеса. Здесь находятся объекты для поддержки процессов управления, обработки счетов, руководства персоналом и т.д. Если моделируется этот уровень, то даже в маленькой компании присутствует большое количество объектов. Следовательно, обычно будет моделироваться только часть из них - та, которая в некотором отношении является критичной.

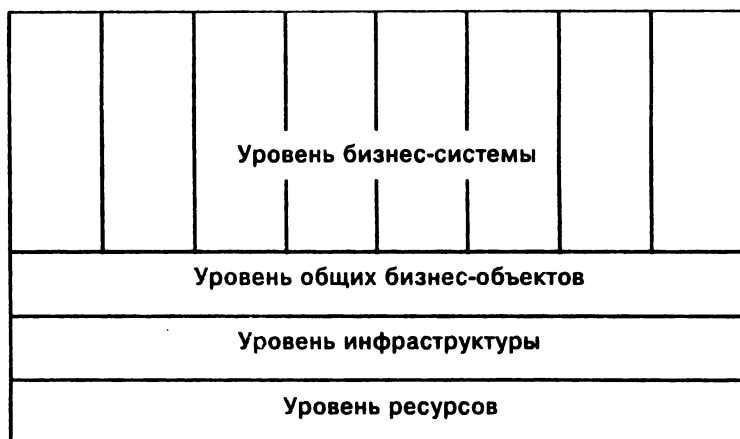


Рис.5.15. Пример многоуровневой архитектуры бизнес-системы

Как владельцы процессов, так и владельцы ресурсов являются исполнителями. Они отчитываются перед президентом моделируемой компании. Владелец ресурса развивает ресурсы, которые могут являться лидерами процессов или операторами процессов. Владелец процесса имеет к лидеру процесса отношение, отличное от традиционного отношения “начальник - работник”. Владелец процесса руководит (тренирует) своих лидеров процессов.

Уровень общих бизнес-объектов содержит объекты, характерные для моделируемого бизнеса. Эти объекты могут использоваться многими бизнесами (экземплярами бизнеса) или сферами бизнеса в моделируемой компании.

Уровень бизнес-системы состоит из объектов, специфичных для данной системы бизнеса. Например, для банка на уровне общих бизнес-объектов будут находиться те объекты, которые используются несколькими банковскими подсистемами, такими, как клиент, счет, кредит, филиал (отделение). Различными банковскими подсистемами

могут быть, например, операции, связанные с депозитированием денег, операциями по ссудам, и операции, связанные с переводом денег из различных частей мира. Они специфичны для моделируемого банка. Бизнес-процессы банка будут видимы в основном на самом верхнем уровне. Именно этот уровень отражает различия между разными банками. Уровень общих бизнес-объектов может быть в большой степени общим для нескольких банков. Другой бизнес, например страховая компания, сможет использовать тот же самый уровень общих объектов, что и банк, но более высокие уровни будут существенно различаться.

Многоуровневая архитектура разрабатывается, как правило, в течение долгого периода времени - итеративно (посредством приближений) или инкрементно (поэтапно). Чтобы упростить понимание, приведем упрощенное представление подхода. Многоуровневая система должна моделироваться как сверху вниз, так и снизу вверх. На самом деле, возможно, правильной будет сказать, что вы моделируете сначала снизу вверх, чтобы знать, к чему привязать вашу модель, построенную сверху вниз.

Можно выделить три этапа:

Э т а п 1. Создать бизнес-систему.

Создать П-модель всей бизнес-системы. В этой модели субъекты будут действующими лицами бизнес-системы, а прецедентами - бизнес-процессы компании. Создать О-модель, соответствующую П-модели.

Э т а п 2. Рекурсивно обработать следующий уровень.

Создать П-модель бизнес-системы, соответствующую следующему уровню и всем его подчиненным уровням. Субъекты этой П-модели берутся из объектной модели этапа 1. Заметим, что не следует путать эти субъекты с субъектами информационной системы, которые рассматриваются в гл. 8. Данные субъекты подобны субъектам, которые моделируют сферы бизнеса (см. подразд. 5.7.2). Создать О-модель для этой П-модели.

Э т а п 3. Продолжить аналогичные действия для каждого уровня.

В системах с многоуровневой архитектурой важно определить интерфейс между разными уровнями. Объектно-ориентированная технология представляет для этого элегантный способ. Каждый уровень - это составной (сложный) объект. Интерфейс для уровня состоит из интерфейсов его публичных (public) объектов. То, как некоторый уровень может использоваться более высокими уровнями, является важным архитектурным решением. Можно выбирать из нескольких видов отношений между уровнями. Рассмотрим два из них:

❶ **Использование объектов нижнего уровня инкапсулировано в объектах верхнего уровня.** При проектировании более верхних уровней интерфейс уровня базовых объектов используют как примитивы, так как интерфейс инкапсулирован в объекты покрывающего уровня и виден только в отношении реализации этих объектов. Интерфейс не

должен использоваться в документации, которая описывает взаимодействие между объектами верхних уровней. Крупная бизнес-система состоит из тысяч объектов, которые общаются друг с другом очень интенсивно. Возможность скрыть коммуникацию с нижележащими объектами упрощает документацию. Так как базовые объекты разрабатываются с перспективой широкого повторного использования, они должны управляться иначе, чем бизнес-объекты: их необходимо тщательно проверить перед реализацией, их нельзя изменять, пока не изменились все зависимые от них бизнес-объекты.

②. *Объекты нижнего уровня должны быть не только инкапсулированы в объектах верхнего уровня, но могут использоваться и в проектной документации верхнего уровня, однако проектировщики верхнего уровня не могут их менять.* Уровень общих бизнес-объектов растет и развивается по мере разработки различных подсистем бизнеса. Он содержит объекты, общие для всего бизнеса и, следовательно, должен стать относительно стабильным после выполнения нескольких версий. Этим уровнем следует управлять не так, как уровнем бизнес-системы, а примерно так, как уровнем инфраструктуры. Важное отличие от объектов на уровне инфраструктуры заключается в том, что при описании подсистем бизнеса необходимо показать объекты уровня общих объектов. Просто использовать эти объекты недостаточно. При разработке подсистем бизнеса можно показать используемые общие бизнес-объекты, но нельзя вносить в них какие бы то ни было изменения. Можно изменять только свои собственные объекты, т. е. объекты, реализующие данную бизнес-подсистему. Если должен измениться общий бизнес-объект, то все бизнес-подсистемы, использующие этот объект, должны быть одновременно изменены.

Каждый уровень может быть разделен на подсистемы, используя методику, описанную ранее в этой главе. Подсистемы на самом верхнем уровне сравнительно велики и независимы друг от друга. Они также могут быть сферами бизнеса, как описано в подразд. 5.7.2.

Сферы бизнеса и уровни взаимно дополняют друг друга. Для описания сложной компании можно применять обе методики. Сначала применяется методика сфер бизнеса для определения различных отраслей, а затем разрабатывается многоуровневая архитектура. Уровни ниже самого верхнего могут использоваться всеми сферами бизнеса.

ЛИТЕРАТУРА

1. Davenport T. H. **Business Innovation, Reengineering Work through Information Technology.** - Boston, MA: Harvard Business School Press, 1993.
2. Hammer M. and Champy J. **Reengineering the Corporation: A Manifesto for Business Revolution.** - N-Y: HarperCollins, 1993.

3. Jacobson I. **Object-oriented development in an industrial environment.** Proceedings of OOPSLA'87. SIGPLAN Notices, 1987, 22(12).
4. Jacobson I., Christerson M., Jonsson P. and Overgaard G. **Object-Oriented Software Engineering - A Use Case Driven Approach.** - Reading, MA: Addison-Wesley; ACM Press, 1992.
5. Jacobson I., Ericsson M., Jacobson A. **The Object Advantage. Business Process Reengineering With Object Technology.** - N-Y: Addison-Wesley Publishing Company, 1995.
6. Martin J. **Enterprise Engineering - The Key to Corporate Survival.** - V. I-V. - UK: Savant Institute. 1994.
7. Meyer B. **Object-oriented Software Construction.** - N-Y: Englewood Cliffs, Prentice Hall, 1988.
8. Shlaer S. and Mellor S.J. **Object-Oriented Systems Analysis.** - Englewood Cliffs, N-Y: Prentice Hall, 1988
9. Sudkamp T.A. **Languages and Machines: An Introduction to the Theory of Computer Science.** - Reading, MA: Addison-Wesley, 1988
10. Wirfs-Brock R., Wilkerson B. and Wienter L. **Designing Object-Oriented Software.** - Englewood Cliffs, N-Y: Prentice Hall, 1990.

МОДЕЛЬ СУЩЕСТВУЮЩЕГО БИЗНЕСА

6.1

Назначение обратного инжиниринга

При разработке образа будущего на основании директив реинжиниринга конкретизируется цель деятельности компании. Для выполнения этой задачи требуется понимание того, как функционирует бизнес в настоящее время. Иными словами, для того чтобы что-то изменить, надо знать текущее состояние. Работу по созданию модели существующей компании называют *обратным инжинирингом*. Результаты этой работы в дальнейшем используются в качестве основы для выполнения прямого инжиниринга компании.

Следует подчеркнуть, что обратный инжиниринг компании имеет самостоятельную ценность вне зависимости от реинжиниринга компании, т.е. от стремления перестроить ее деятельность. Действительно, даже если у руководства компании нет потребности/желания осуществлять реинжиниринг, может существовать серьезная потребность в создании модели существующего бизнеса, которая позволяет как вскрыть существующие резервы компании, так и показать ее узкие места. Для иллюстрации самостоятельной ценности проведения обратного реинжиниринга, в Приложении 2 описан опыт создания модели существующего бизнеса АО “Шереметьево-Карго”, предпринятый с целью проведения аналитических исследований эффективности функционирования его подразделений [1,2]. Эта модель позволяет:

- Предсказать последствия разнообразных внешних воздействий на компанию: например, “Может ли компания перевезти дополнительное количество рейсов (к существующему расписанию)?”, “Если нет, то что этому препятствует (недостаток тракторов, грузчиков, складов) и т.д.”.

- Вычислить графики грузопотоков, удельной загрузки складов, коэффициентов использования тракторов (тележек), графики экономических показателей и т.д.

Заметим, что многие компании, поняв актуальность и важность БПР, на первом этапе, (если им не грозит крах в ближайшее время) ограничиваются только разработкой модели существующего бизнеса и только затем осуществляют проектирование нового бизнеса.

Обратим внимание на то, что описание модели АО “Шереметьево-Карго” выполнено на базе интегрированного многофункционального динамического средства ReThink (см. разд. 9.3), что позволило избежать ограничений статических моделей, указанных ранее в п. 5.1.

Обратный инжиниринг состоит из двух основных этапов:

① Моделирование на основе прецедентов (П-модель) используется для создания и описания модели существующего бизнеса в терминах субъектов и прецедентов. Такая модель является внешним представлением компании. Она используется для определения приоритетных процессов (прецедентов), подлежащих реинжинирингу. В ней описываются потоки событий в приоритетных прецедентах. Для каждого прецедента строится метрика. Полученные метрики в дальнейшем используются для определения того, достигаются ли поставленные цели в новом бизнесе.

② Объектное моделирование (О-модель) осуществляется для создания объектной модели существующего бизнеса. Чтобы не увязнуть в излишней детализации существующего бизнеса, на этом этапе следует моделировать только функциональную иерархию компании и то, как прецеденты распределяются через подсистемы, соответствующие функциональным подразделениям компании.

При разработке образа будущего создается первая грубая модель бизнес-процессов, которую предполагается улучшить в ходе реинжиниринга. Результат используется в качестве входа для обратного инжиниринга. П- и О-моделирование зависят друг от друга. Сначала строится П-модель. Когда она становится стабильной, то начинают объектное моделирование. Иногда работу по идентификации некоторых очевидных объектов можно вести параллельно с П-моделированием, однако чаще всего хороший результат не может быть получен без итераций между моделями (рис. 6.1).

Результатом обратного инжиниринга является модель текущего состояния бизнеса, представленная П-моделью и О-моделью. На практике при выполнении обратного инжиниринга целесообразно использовать только те средства, которые помогают в достижении наилучшего результата, и отбросить те, которые, не давая улучшений, только замедляют работу. В следующих разделах будут даны советы, как использовать только то, что нужно.

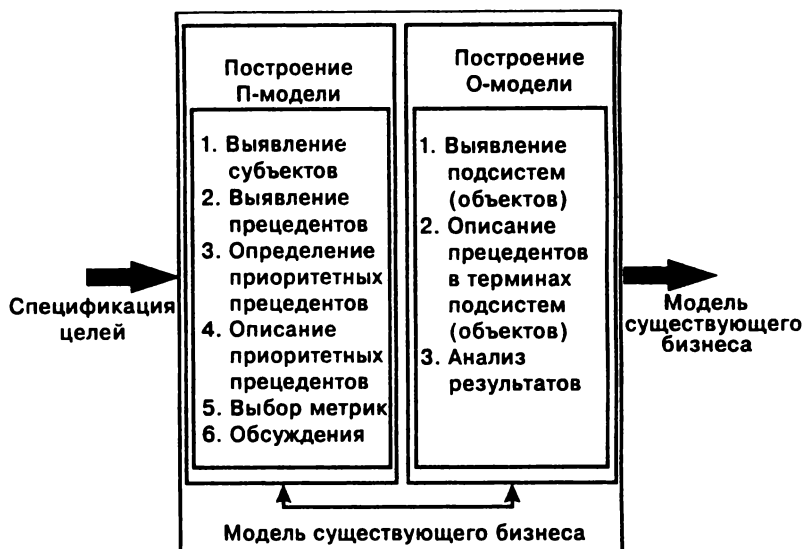


Рис.6.1. Обратный инжиниринг

Не всегда бывает так, что компания, желающая провести реинжиниринг своих процессов, имеет хорошо документированную модель существующего бизнеса. Даже если такая модель существует, она очень редко описывает действительное состояние дел. Еще менее вероятно, что эту модель смогут понять те, кто отвечает за бизнес, и те, кто выполняет реинжиниринг. Причин тому несколько. Во-первых, люди достаточно заняты собственно бизнесом, и считается, что на создание модели или ее обновление нет времени. Во-вторых, считается, что бизнес - это слишком сложный предмет, и его нельзя хорошо описать. Кроме того, традиционные методы описания бизнеса понятны только тем, кто эти описания создает, но не самим предпринимателям; способ описания не интуитивен и не очевиден, часто для его понимания требуется владение определенной сложной методикой. Тех, кого интересует, как же в действительности происходит моделирование бизнеса традиционными методами, можно отослать к разд. 3.4.

Так стоит ли, учитывая сказанное выше, создавать модель существующего бизнеса? Некоторые авторы полагают, что лучше начать все с нуля, чем пытаться понять существующее положение дел. Люди опасаются, что несовершенные знания о том, как компания работает сейчас, могут ограничить творчество и, как следствие, ограничить возможные перемены. Здесь мы приходим к классической проблеме: может ли знание быть опасным? Мы предполагаем, что люди, проводящие реинжиниринг, смогут абстрагироваться от существующей

структуры бизнеса и думать о том, как бизнес должен работать завтра. Специалисты должны понимать, что если они ограничивают полет своей фантазии, то не смогут произвести изменения, необходимые для того, чтобы сделать компанию конкурентоспособной. Конечно, можно предложить изменения в бизнесе без проведения обратного инжиниринга, но мы полагаем, что без адекватной картины текущего положения дел в компании не удастся найти лучшие пути для ее изменения. Если участники команды по реинжинирингу поймут нынешнюю ситуацию в бизнесе, то они смогут избежать неэффективных путей при изменении этого бизнеса. Таким образом, команда по реинжинирингу завоюет доверие как рядовых работников, так и руководства, и ее предложения будут встречать поддержку. Не следует легкомысленно относиться к последнему утверждению.

Привлечь персонал на свою сторону - возможно, самая тяжелая задача при проведении реинжиниринга, так как в результате повседневная работа служащих может измениться до неузнаваемости.

Создать простое, ясное и достаточно общее описание бизнеса вполне возможно. При этом важно иметь как общие описания, дающие в целом картину бизнеса, так и конкретные описания. Каждое конкретное описание обрисовывает свою часть бизнеса или его процессов, но при необходимости отдельные описания должны объединяться в общее. Важно предоставить каждому сотруднику возможность принимать участие в составлении моделей. Люди могут быть заинтересованы в рассмотрении бизнеса на разных уровнях детализации в зависимости от тех ролей, которые эти люди играют в организации. Более общие описания могут быть интересны руководству, которое обычно не интересуется деталями. Менеджерам процессов, наоборот, требуется понимание всех деталей тех процессов, за которые они отвечают, и иногда соседних. Необходимо понимать существующий бизнес, как минимум, в тех областях, где предстоит проводить изменения. Это необходимо, так как позволит:

- людям, выполняющим работу по реинжинирингу, понять недостатки существующего бизнеса и определить те области, в которых стоит производить изменения;
- понять, как следует изменить бизнес, чтобы он удовлетворял новым требованиям и целям;
- объяснить сотрудникам, чем плохи прежние процедуры работы и почему они должны быть приспособлены к новым способам работы;
- измерить характеристики тех прецедентов, которые следует изменить. Полученные измерения затем можно будет использовать для оценки экономического эффекта проведенных изменений.

6.2

Общая картина бизнеса

Определив и описав процессы, обеспечивающие наибольшие улучшения, можно сосредоточиться на собственно реинжиниринге. С этого момента начинают проводить измерения параметров выделенных приоритетных процессов. Полученные значения используют для подтверждения, что новые процессы приведут к кардинальным изменениям.

Обычно удобно начинать с составления обзорного плана всего бизнеса, а не только приоритетных процессов. При этом достигается экономия времени, если все участники реинжиниринга согласны с общей картиной бизнеса. Создание такой модели может затягиваться, особенно при выполнении первого проекта по реинжинирингу, поэтому необходимо идти на компромисс и описывать только процессы с наивысшими приоритетами, не принимая во внимание остальные.

Одна из сложнейших задач при описании бизнеса - найти правильный уровень детализации. Не следует моделировать только ради моделирования: модели должны помогать, а не мешать! Метод, применяемый для моделирования бизнеса, должен предлагать архитектуру, способную описывать бизнес как на высоком уровне абстракции, так и в деталях там, где это нужно. Как правило, достаточно идентифицировать процессы и построить на их основе описание высокого уровня, не вдаваясь в подробности. Если показать, как приоритетные процессы пронизывают существующие функциональные единицы бизнеса, то можно получить достаточно хорошее понимание для моделирования нового бизнеса. Обычно рекомендуется ограничивать время создания существующей модели; для компании среднего размера это должны быть недели, а не месяцы.

6.3

П-модель существующего бизнеса

Вначале создается общая П-модель, обрисовывающая весь бизнес, в ней определяются приоритетные прецеденты, затем эта общая модель используется для создания более детальной. Таким образом вырабатывается общая картина бизнеса, которую можно сообщить каждому сотруднику и обсудить с руководством. Вся компания в целом редко подвергается реинжинирингу. Сначала предпочтение отдается самым неотложным прецедентам; они получают наивысший приоритет, и для них строятся полные модели и описания. Работа

над созданием П-модели часто начинается с “мозгового штурма”, в ходе которого команда по реинжинирингу набрасывает первоначальный набор субъектов и прецедентов существующего бизнеса.

Рассмотрим процесс построения полной П-модели, не зависящий от того, общая или более детальная модель требуется.

6.3.1

Выявление субъектов

Создание П-модели рекомендуется начинать с определения субъектов бизнеса. Наиболее важными субъектами являются клиенты компании. Кроме них, типичными субъектами обычно являются партнеры, поставщики и филиалы. Если моделировать только часть компании, то остальные части компании будут выступать в роли субъектов. Это означает, что даже отдельные люди в других отделах компании станут субъектами. Нечто является субъектом бизнеса, если оно находится вне моделируемой части бизнеса; взаимодействует с этим бизнесом тем или иным способом и если существуют его типичные экземпляры.

Для определения того, все ли субъекты бизнеса найдены, необходимо рассмотреть примеры нескольких реальных людей, имеющих контакты с компанией. Сгруппируйте этих людей в соответствии с ролями, которые они выполняют по отношению к компании. Каждой группе поставьте в соответствие субъект, который следует включить в модель.

6.3.2

Выявление прецедентов

Когда все члены рабочей группы согласны с полученным в первом приближении перечнем субъектов, можно начинать определение набора прецедентов бизнеса. Начните с субъектов-клиентов. Не увлекайтесь излишней детализацией. Помните, что прецеденты моделируют завершённые потоки событий и должны иметь измеримую ценность для клиента (покупателя).

На ранней стадии моделирования можно рассматривать различные варианты модели, приближаясь к такой, которая всех устроит. Чтобы убедиться в том, что все аспекты (обязательства) бизнеса действительно включены в П-модель, необходимо, чтобы каждому обязательству соответствовал хотя бы один прецедент.

Определение подходящих прецедентов может быть нелегкой задачей. Границы между функциональными единицами подразделения иногда кажутся непреодолимыми преградами, и часто не очень по-

нятно, как прецеденты их пересекают. Не следует выделять слишком много прецедентов бизнеса. Правильно, если их будет от 10 до 20. Если прецедентов бизнеса больше тридцати, то вы, вероятно, забыли, что прецеденты - это завершённые потоки событий.

Рассмотрим на примере некоторой абстрактной компании типичные прецеденты бизнеса (рис. 6.2). Во многих компаниях можно выделить следующие типы субъектов. Клиент (Покупатель), моделирующий людей, потенциально имеющих или могущих проявить интерес к покупке продуктов; Пользователь, моделирующий людей,

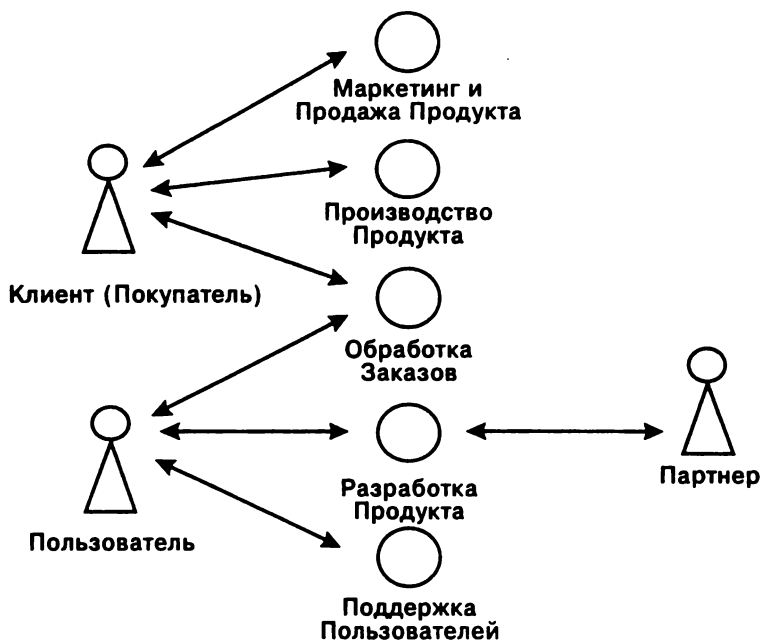


Рис.6.2. П-модель существующего бизнеса

для которых в настоящий момент производится продукция; Партнер, который моделирует любую компанию, способствующую разработке продуктов. Обычно прецедентами являются: Маркетинг и Продажа Продукта - процесс, способствующий распространению информации о продукте на рынке; Обработка Заказов - процесс, состоящий в регистрации и обработке приходящих заказов; Разработка Продукта - процесс, осуществляющий проектирование продукта; Поддержка пользователей - процесс, обеспечивающий пользователей интересующей их информацией и т.п.; и Производство - процесс, осуществляющий изготовление (тиражирование) продукта.

При внимательном анализе идентифицированных прецедентов могут выявиться некоторые неопределенности. Может оказаться, что определенные прецеденты не связаны ни с одним субъектом. Конечно, подозрительно, что эти прецеденты никого не интересуют и никем не используются. Могут ли такие прецеденты существовать? Да, и такое происходит в том случае, если эти прецеденты представляют собой такие потоки событий, содержание которых зависит от того, как реализуется бизнес. Подобное наблюдается и для потоков, состоящих из действий, необходимых для обеспечения нормальной работы компании. Как правило, это потоки событий, поддерживающие инфраструктуру бизнеса в компании. Если это так, то такие прецеденты следует рассматривать на более низком уровне детализации, отражающем внутреннюю структуру бизнеса. Подобные прецеденты могут также быть только фрагментом потока событий, и их выделили только потому, что они “всегда” считались самостоятельными. Типичный пример такого несамостоятельного прецедента - планирование.

Не следует тратить слишком много времени на построение идеальной картины работы бизнеса; довольствуйтесь такой моделью, которая отражает все важные потоки событий, и переходите к следующему шагу.

Заметим, что описанная ранее методика моделирования (см. гл. 5) позволяет структурировать прецеденты с помощью отношений использования и расширения. Однако при моделировании существующего бизнеса не следует тратить на это слишком много времени. Применение этих отношений для моделирования нового бизнеса будет описано в гл.7.

6.3.3

Определение приоритетных прецедентов

Итак, первый результат работы - предварительная модель субъектов и прецедентов бизнеса. На основе этой модели следует выделить те прецеденты, с которых надо начинать работу по реконструкции. Бизнес-процессам, которые являются критическими для работы команды по реинжинирингу, дается приоритет. К числу приоритетных прецедентов следует отнести те, которые обладают некоторыми из следующих характеристик:

- оказывают наибольшее влияние на клиентов бизнеса;
- с большой вероятностью приведут к радикальным улучшениям;
- имеют проблемы при реализации в существующем бизнесе;
- могут быть легко и быстро улучшены;
- улучшение которых окажет наибольший эффект на внутреннюю структуру самого бизнеса.

Определение приоритетных прецедентов проводится с помощью рангов. Ранг обозначает важность прецедента. Прецедентам, реализующим главные задачи бизнес-системы, назначают наивысший ранг. Проранжируйте прецеденты и проводите дальнейший процесс моделирования в соответствии с полученными рангами. На первом этапе моделируются прецеденты высшего ранга (X); на следующем - ранга (X-1) и т. д.

6.3.4

Описание прецедентов

Каждый приоритетный прецедент модели должен быть полностью описан. Прецедент можно описать неформально. Следует избегать излишней формальности. Одновременно с описанием прецедентов описываются и их субъекты, благодаря чему можно четко определить интерфейс между прецедентами и субъектами.

Описав основной ход событий прецедента, следует описать важнейшие альтернативные или дополнительные потоки событий и исключений. В процессе этой работы следует обращать особое внимание на обстоятельства, которые могут привести к прерыванию основного хода событий и переключению на другую последовательность событий. Вся совокупность альтернативных потоков событий обычно требует более обширной деятельности, чем основной ход событий. Следовательно, для успешного реинжиниринга очень важно, чтобы все эти альтернативы были учтены и проработаны.

Легко увязнуть в работе по описанию прецедентов, слишком сильно их детализируя. Имейте это в виду и старайтесь при описании прецедентов рассматривать их с достаточно абстрактной точки зрения. Не позволяйте этой части работы слишком затянуться и не расходуйте на нее весь запас энтузиазма; он вам еще пригодится для самого реинжиниринга.

Описание прецедента должно включать только то, что осуществляется в нем самом, и взаимодействие данного прецедента с его субъектами. Не надо описывать поведение субъектов применительно к рассматриваемому прецеденту. Таким образом, если границы системы определены нечетко (то есть не определено, что лежит вне бизнеса, а что - внутри), то описание прецедентов может оказаться непрос-той задачей.

Определение границ - один из наиболее важных результатов П-моделирования, ибо если нет ясности в существующем бизнесе, то не ясно как его реконструировать.

6.3.5

Выбор метрик

Метрики - это ключ к увеличению эффективности выполнения прецедента, так как они помогают понять действительные цели реинжиниринга. Как сказал Дж.Харрингтон [1]:

“Измерения - это ключ. Если вы не можете измерить, то не сможете контролировать. Если не сможете контролировать, то не сможете управлять. Если не сможете управлять, то не сможете улучшить. Это так просто!”

Выбрать метрику иногда достаточно сложно. Выбор зависит от прецедента, с которым вы работаете. В хорошем наборе метрик обычно присутствуют метрики времени, цены и качества, причем обычно бывает так, что если усилия сосредотачиваются на улучшении временных показателей (например, времени выполнения) прецедента, то показатели цены и качества улучшаются автоматически.

Выбирая метрики, не забывайте о том, что клиент должен понять и оценить улучшения. Поэтому выбирайте метрики с точки зрения клиента и будьте уверены, что выбранные метрики могут быть измерены. Например, для прецедента Обработка Заказов можно использовать такие метрики:

- средняя стоимость обработки одного заказа;
- среднее время обработки заказа;
- процент отказов.

6.3.6

Обсуждения

Когда описания прецедентов достигли относительно стабильного состояния, следует провести обсуждения (см. разд. 4.8). Важно, чтобы представители всех точек зрения согласились с полученным списком прецедентов. Ниже приведен список вопросов, которые следует обсудить для каждого прецедента.

- Правильно ли описано взаимодействие с субъектом(ами)?
- На подходящем ли уровне детализации описан ход событий?
- Является ли ход событий корректным и завершённым?
- Все ли важные альтернативные потоки событий описаны?

Для выявления всех ошибок на раннем этапе необходимо присутствие на совещании представителей команды по реинжинирингу, сотрудников, выполняющих данный процесс, владельцев и лидеров процессов.

О-модель существующего бизнеса

Как сказано в главе 5, объектная модель представляет собой модель бизнеса в терминах взаимодействующих между собой объектов. Эти объекты группируются в подсистемы, причем прецеденты могут работать с объектами из различных подсистем.

При проведении обратного инжиниринга обычно удобно начинать с определения подсистем и затем описывать прецеденты в терминах общения подсистем друг с другом. Если необходимо, то моделирование подсистем осуществляется в терминах объектов.

6.4.1

Выявление подсистем

Большинство компаний на сегодняшний день имеют функциональную иерархическую структуру, которая может быть промоделирована выделением одной подсистемы на каждую функциональную единицу. В каждой компании эти единицы называются по-своему; чаще всего встречаются такие названия, как отделение, отдел, секция, департамент, лаборатория или группа. Поскольку каждая такая единица может включать в себя более мелкие, то часто (особенно в больших компаниях) существует несколько уровней подсистем.

Рис. 6.3 содержит пример типичной структуры современного бизнеса. Показаны только подсистемы верхнего уровня, но каждая из них может состоять из меньших подсистем. Этот тип отношений называется отношением *“состоит из”*. Количество уровней зависит от компании. Такая модель служит для глубокого понимания прецедентов.

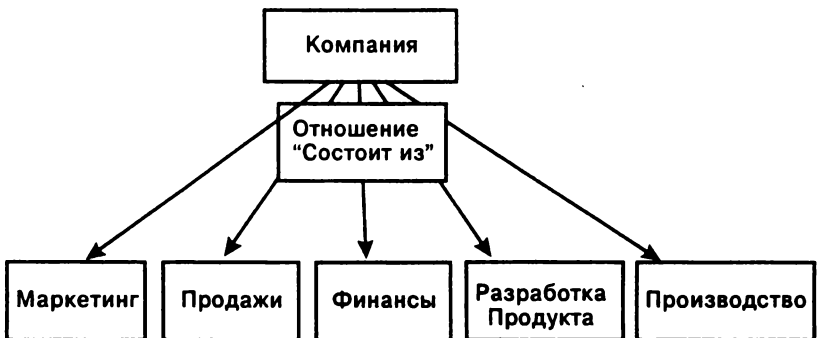


Рис.6.3. Иерархия подсистем в компании

Сфера деятельности каждой функции бизнеса имеет текстовое описание, поэтому механизм взаимодействия организационных единиц обычно вполне понятен.

6.4.2

Описание прецедентов в терминах взаимодействующих подсистем

Определив подсистемы, следует заново описать каждый прецедент, но на этот раз в терминах взаимодействующих подсистем. Необходимо понимать, какие подсистемы участвуют в потоке событий для каждого прецедента. Это проясняет взаимодействие подсистем друг с другом и показывает, какие подсистемы должны работать вместе в каждом конкретном прецеденте.

Определение прецедента в терминах сотрудничающих подсистем может быть описано при помощи диаграмм взаимодействия прецедентов, рассмотренных в подразд. 5.4.2. Однако в большинстве случаев этого не требуется. Вместо этого достаточно указать, что, например, прецедент Маркетинг и Продажа Продукта выполняется подсистемами Маркетинг, Продажи и Разработка Продукции.

6.4.3

Выявление объектов

В большинстве случаев достаточно определить подсистемы и описать прецеденты в терминах взаимодействующих подсистем. Иногда, конечно, приходится проводить более детальное рассмотрение, но в этом случае необходимо оценить возможные негативные последствия. Последствия могут быть такими:

- Задержка проекта из-за углубления в детали.
- Из-за длительности разработки участники команды по реинжинирингу, а также и прочий персонал компании могут потерять веру в то, что произойдут положительные изменения.
- За время разработки деталей может измениться сам бизнес.

Если все же необходимо иметь более подробную модель существующего бизнеса до начала разработки нового, можно на основе прецедентов определить объекты и затем показать, как прецеденты реализуются в виде взаимодействия этих объектов. В этом случае для каждого прецедента строится представление на основе сотрудничающих объектов, показывающее, как эти объекты, работая совместно, реализуют прецедент. Для демонстрации того, как объекты сотрудничают, можно воспользоваться диаграммами взаимодействия. Определение объектов на основе прецедентов и методика построения диаграмм взаимодействий более подробно описаны в гл. 7.

Анализ результатов

Когда команда по реинжинирингу достигла согласия во взглядах на то, как протекают события в приоритетных прецедентах, следует провести оценку каждого из них. Насколько глубоко проводить анализ - дело разработчиков, но обязательно надо выполнить следующие действия.

❶ Измерить параметры существующих прецедентов в соответствии с выбранными метриками. Полученные данные будут впоследствии сравняться со значениями модифицированных прецедентов. Если цель изменения прецедента состоит в сокращении времени выполнения, то важно иметь возможность измерить это сокращение, то есть убедиться, что цель достигнута.

❷ “Прогнать” прецеденты по одному и критически рассмотреть ход событий. Классифицировать каждое отдельное действие прецедента как “Увеличивающее потребительскую Ценность продукции” (УЦ) или “Не Увеличивающее Ценность” (НУЦ). С точки зрения клиента, УЦ-действие приводит к увеличению ценности конечной продукции. После проведения классификации для каждого НУЦ-действия определяется действительная его необходимость и стоимость выполнения. Следует рассмотреть возможность уменьшения стоимости выполнения УЦ-действий и, по возможности, устранения НУЦ-действий. Некоторые НУЦ-действия не могут быть устранены; например, если они предписаны законодательством. Такие “прогоны” прецедентов дают хорошую почву для оптимизации процесса.

❸ Основываясь на значении метрик существующих прецедентов, можно идентифицировать проблемы и ограничения, связанные с ними. Для получения такой информации хорошие результаты обычно дает опрос персонала, связанного с рассматриваемыми прецедентами. Опрос проводится для того, чтобы понять проблемы, связанные с существующими прецедентами, и получить предложения по их усовершенствованию. Если удастся заинтересовать людей в работе над усовершенствованиями, вы с удивлением обнаружите, что они могут предлагать конструктивные идеи.

❹ Исследовать краткосрочные рационализаторские предложения по улучшению существующих прецедентов. Чтобы обнаружить пути более радикальных изменений, сначала надо понять, что можно сделать простым улучшением существующего бизнеса. Важно также, чтобы персонал видел, что уже на раннем этапе работы по обновлению происходит прогресс, что повышает его мотивацию. Для этого

начальные работы должны легко реализовываться и быстро давать желаемый результат. С точки зрения персонала даже небольшие изменения в повседневной работе могут рассматриваться как нечто очень позитивное.

5 Определить существующую информационную поддержку. Определить прецеденты, использующие компьютерную поддержку. Описать, как ИТ оптимизируют работу, и обсудить достоинства и недостатки ее использования. Исследовать, существует ли возможность увеличить эффективность прецедентов, использующих компьютерную поддержку. Например, для тех задач, которые раньше выполнялись вручную, можно попытаться применить компьютер. Иногда просто невозможно реконструировать бизнес, не построив совершенно новую систему компьютерной поддержки. В то же время важно осознавать, что нельзя одним махом выкинуть все, во что вложены деньги, зачастую немалые. Поэтому вместе с определением приоритетных процессов бизнеса необходимо определить и приоритетные процессы в создании компьютерной поддержки (см. более подробно гл.8).

6 Провести “инвентаризацию” существующих знаний и областей компетенции сотрудников. Это необходимо для понимания того, какие дополнительные навыки и знания потребуются от персонала в новом бизнесе. Часто бывает полезно рассмотреть роли, которые играют в существующем бизнесе основные сотрудники для того, чтобы в случае необходимости иметь возможность распространить их знания и, возможно, разделить их работу между несколькими специалистами. Основные люди есть в любой компании. Это те, кто управляет развитием компании или обладает уникальными знаниями о ее продукции. Они, конечно, хорошие специалисты, но зависеть от нескольких личностей очень опасно, так как нет никаких гарантий, что компания всегда сможет обеспечить наличие этих людей. Если такой человек заболеет или покинет компанию, это может привести к катастрофе.

7 Попытаться определить требования к будущим изменениям прецедентов. В результате может оказаться, что бизнес должен предложить совершенно новые прецеденты, которые раньше даже и не рассматривались. Такое происходит либо потому, что конкуренты уже предложили определенные услуги, которых требуют клиенты, либо из-за того, что компания нашла способ увеличения своей конкурентоспособности. В первом случае требуется предложить нечто, что по крайней мере не хуже, чем продукция конкурентов. Во втором случае положение компании гораздо выгоднее, так как она опережает конкурентов, но успех требуется закреплять.

ЛИТЕРАТУРА

1. Кисель Е.Б., Шинкарев М.Б. Опыт использования системы **ReThink** в АО **Шереметьево-Карго**. Материалы семинара "Бизнес-процесс реинжиниринг и проектирование информационных систем". Москва, МЭСИ, 1996.
2. Попов Э.В., Фоминых И.Б., Кисель Е.Б., Шапот М.Д. Статические и динамические экспертные системы. М: Финансы и статистика, 1997.
3. Harrington J. **Business Process Improvement**. N-Y: McGraw-Hill, 1991.

7

МОДЕЛЬ НОВОГО БИЗНЕСА



7.1

П-модели нового бизнеса

В процессе работы над созданием схемы будущей компании определяются процессы, подлежащие реинжинирингу. Цели нового бизнеса компании выражаются в терминах этих процессов в спецификации целей. В начале прямого реинжиниринга следует выяснить, как новые цели повлияют на существующий бизнес, и определить наилучший способ улучшения и оптимизации выбранных ранее приоритетных процессов.

Инжиниринг нового бизнеса включает пять основных этапов.

- ① *Создание П-модели*, т.е. “внешнего” представления новой компании. Прецеденты при этом отражают процессы нового бизнеса.
- ② *Построение идеальной О-модели* “внутреннего” представления компании, описывающей компанию в терминах общающихся объектов и отражающей то, как эти объекты взаимодействуют, реализуя прецеденты.
- ③ *Создание реальной О-модели*, адаптирующей идеальную модель к ограничениям конкретного бизнеса и разработка реальной объектной модели бизнеса.
- ④ *Разработка информационной системы*, осуществляющей поддержку нового бизнеса с помощью ИТ. Этот этап может оказаться довольно трудоемким.
- ⑤ *Тестирование нового бизнеса*. До того, как нововведения будут внедрены во всей компании, их следует проверить в меньшем масштабе.

Хотя все этапы взаимосвязаны, работы в основном должны выполняться параллельно. Тем не менее, общая последовательность работ при реинжиниринге направлена слева направо (рис. 7.1). Чтобы получить наилучший результат, работу необходимо проводить итеративно.

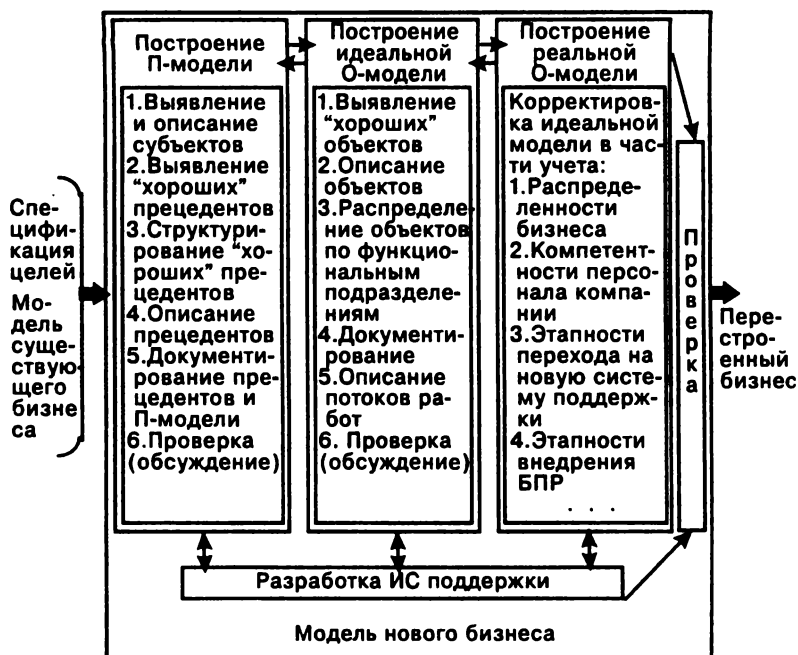


Рис.7.1. Прямой инжиниринг

Построение П-модели нового бизнеса осуществляется так же, как описано в гл. 6. Исходными данными для нее являются результаты работы по разработке образа новой компании (спецификация целей компании) и работы по обратному инжинирингу (модель/модели существующего бизнеса). Рассмотрим подробнее этап построения П-модели нового бизнеса.

7.1.1

Определение и описание субъектов

Основываясь на модели субъектов существующего бизнеса и спецификации целей компании, необходимо определить, как новые требования могут повлиять на эту модель. Для каждого субъекта следует проверить, не изменилась ли его роль при определении новых целей бизнеса, так как иногда в результате изменения бизнеса оказывается возможным удалить некоторый субъект, присутствовавший в модели существующего бизнеса. Например, компания может больше не нуждаться в поставщике какого-либо компонента продукции, а начать производить его самостоятельно. Кроме того, могут появиться

ся совершенно новые субъекты. Такое может случиться, если компания решила не выполнять некоторую задачу самостоятельно, а воспользоваться услугами специализированной компании.

Часто бывает и так, что субъект имеет в новом бизнесе совершенно другую роль: определенные задачи, которые раньше выполнял этот субъект, теперь выполняет компания и наоборот. Конечно, существуют и такие субъекты, которых перемены не затрагивают.

Изменения в модели субъектов могут повлиять на их описание. В этом случае требуется описать новых субъектов и модифицировать описания тех, чьи роли по отношению к компании изменились. Как правило, субъекты описываются очень кратко. Важно, чтобы роли субъектов были описаны четко. Как упоминалось выше, субъекты определяют, где проходят границы бизнеса. В начале моделирования точно определить эти границы (то есть указать, что лежит вне, а что внутри системы) обычно достаточно сложно, но в процессе работы эти границы становятся все конкретнее, поэтому описания субъектов приходится моделировать по несколько раз. Важно осознавать преимущества четко определенных субъектов, так как, если границы не четко определены, то невозможно корректно описать прецеденты. Кроме указания роли субъекта, описание должно содержать информацию о тех прецедентах, с которыми субъект взаимодействует, и то, как он с ними взаимодействует.

7.1.2

Определение и описание прецедентов

На этом этапе исследуется влияние спецификации целей компании, выраженной в терминах процессов, на существующую П-модель. Каждое существенное изменение модели скорее всего означает, что требуется новое понимание прецедентов в целом. Такие перемены влекут за собой необходимость интерактивного рассмотрения различных возможных П-моделей в поисках той, которая наилучшим образом отражает новые цели. Обычно эта задача решается методом “мозгового штурма” и, если возможно, с использованием прототипов (разд. 7.5). Обычно радикальные изменения происходят в рамках существующих прецедентов, но вполне возможно, что новые требования окажутся настолько радикальными, что потребуются ввести совершенно новый прецедент.

Модификацию П-модели полезно проводить группой, причем в состав группы должны входить участники команды по реинжинирингу и участники тех частей бизнеса, которые описываются в данной П-модели. Полезным окажется присутствие человека с опытом проведения других проектов по реинжинирингу. Если такого нет в коллективе компании, можно пригласить консультанта со стороны. Однако ответственность лежит на руководстве команды по реинжинирингу.

Что такое хороший прецедент. Одна из самых важных характеристик прецедента - его способность отражать “хорошие” бизнес-процессы. Понятие “прецедент” точнее, чем “процесс”. В определении прецедента четко указано, как хочет его видеть клиент, а не то, как участники бизнеса его структурируют. Более формальное описание прецедента приведено в гл. 5. Приведем несколько рекомендаций по составлению “хороших” прецедентов.

Некоторые команды по реинжинирингу считают, что необходимы подробные модели с большим количеством маленьких прецедентов. Существуют случаи, когда для адекватного представления бизнеса использовались сотни прецедентов. Другие команды, наоборот, полагают, что в модели должно быть два, в крайнем случае три прецедента. Дж.Рокарт и Дж.Шорт [5], например, определили три основных процесса: разработка новой продукции, поставка ее клиентам и управление отношениями с клиентами. Для определения нужного количества прецедентов требуется достигнуть компромисса между простотой и понятностью картины бизнеса, с одной стороны, и сложностью понимания, измерения и модификации отдельного прецедента, с другой стороны. Чем крупнее прецеденты, тем проще рассмотреть бизнес в целом, но в этом случае, к сожалению, отдельный прецедент становится слишком абстрактным и плохо отражающим существо дела.

Как правило, компромисс найти удастся. Большинство компаний можно достаточно подробно описать при помощи примерно 15 основных прецедентов. Сами прецеденты при этом достаточно невелики, поэтому их можно легко понимать, обрабатывать и модифицировать. С другой стороны, этого вполне достаточно для того, чтобы иметь исчерпывающую картину работы бизнеса. Т.Давенпорт [1] также предлагает использовать 10 - 20 прецедентов и применять методику Дж.Харрингтона [3], согласно которой руководство компании определяет, кто и за какие прецеденты отвечает, и каким образом конструируется цельная картина.

Перечислим некоторые общие характеристики “хорошего” прецедента.

- Прецедент должен быть четко определен и прост для понимания. Важно, чтобы было легко интерпретировать, какому описанию прецедента следовать при получении бизнесом того или иного стимула.

- Следует стремиться к тому, чтобы в выполнении экземпляра прецедента участвовало как можно меньше людей. Важнее всего, чтобы интерфейс с клиентом был настолько простым, насколько это возможно. Если прецедент достаточно мал, то желательно выполнять его одним человеком. М.Хаммер и Дж.Чампи [2] называют такого человека “исполнитель прецедента”. Наличие такого человека уменьшает количество передач дела из рук в руки, при которых возникают задержки, за счет чего временные показатели ухудшаются.

- Все шаги прецедента должны по возможности способствовать увеличению ценности продукта для клиента.

- Люди, реализующие задачи прецедента, должны обладать необходимыми полномочиями. Эти люди не должны ждать решения руководства, если произойдет что-либо непредусмотренное: у них есть полномочия принимать решения самостоятельно.

- Должен существовать простой способ адаптации прецедента к любым ограничениям бизнеса. Прецедент или, точнее, его класс, может существовать в нескольких вариантах. Например, многонациональная компания может иметь в разных странах похожие, но имеющие определенные отличия процессы продаж. Различные компании могут иметь похожие процессы разработки продукта, но эти процессы будут тем не менее отличаться друг от друга в зависимости от размера бизнеса, вида продукции, уровня подготовки персонала и т.д.

Структурирование прецедентов. После начального определения прецедентов необходимо структурировать П-модель. Для этого существуют различные методики.

Один способ - применить отношение использования (см. гл. 5). Если два (или более) конкретных прецедента имеют достаточно похожее поведение, то следует выделить общую часть в отдельный прецедент и установить между ним и исходными отношение использования. В этом случае общее поведение описывается только один раз, что значительно упрощает будущие изменения.

Другой способ - использование отношения расширения. В ряде случаев целесообразно извлечь из прецедента описание некоторого фрагмента событий и поместить это описание в отдельный прецедент, имеющий отношения расширения с исходным. Перечислим эти случаи.

- Фрагмент потока событий рассматривается как очень важный и существенный. В результате извлечения этот фрагмент становится ясно виден на общей схеме прецедентов.

- Поведение фрагмента потока событий сильно отличается от поведения основного потока событий, то есть фрагмент содержит совершенно отличные от основного потока событий работы и обязательства. Другая причина выделения фрагмента потока событий может заключаться в том, что он имеет периодичность, несколько отличающуюся от периодичности основного потока.

- Фрагмент потока событий встречается в нескольких прецедентах. Вместо описания его в нескольких прецедентах как альтернативный поток такой фрагмент можно описать только один раз. (В этом случае можно применять и отношение использования, но при использовании отношения расширения исходному прецеденту не надо знать, куда и когда извлеченный прецедент будет вставлен).

Типы прецедентов, созданных при помощи отношения расширения и использования, должны быть достаточно существенны сами по себе, иначе эффекта не будет. Однако в каждом прецеденте обычно есть один или несколько альтернативных потоков событий. Их следует отражать не созданием новых прецедентов, а структурированием описания существующего. В результате эти потоки событий явно не видны в П-модели.

Теперь приведем пример различных способов структурирования прецедентов. Рассмотрим подробнее, как можно структурировать прецедент Продажа. Если клиент - это человек, желающий купить стандартный продукт, то Продажа выглядит совершенно иначе, чем в случае, когда клиент желает приобрести заказной продукт. Индивидуальный клиент, заинтересованный в покупке стандартного продукта, не должен проходить сложную процедуру, существующую при покупке заказного продукта: он должен быстро получить то, что ему надо. Как было описано выше, это можно промоделировать по-разному. Выбор альтернатив зависит от требований, выдвигаемых к модели. Начнем с применения отношения использования (рис. 7.2). Рисунок означает, что клиент инициирует либо Продажу Готового типового Продукта, либо Продажу Заказного Продукта. Продажа Готового Продукта производится, когда клиент покупает продукт либо прямо со склада, посетив компанию, либо по почте. Продажа Заказного Продукта означает, что с отделом продаж компании связывается представитель другой фирмы и обсуждает возможность заказа крупной партии специально разработанной продукции. Продажа Готового Продукта и Продажа Заказного Продукта содержат

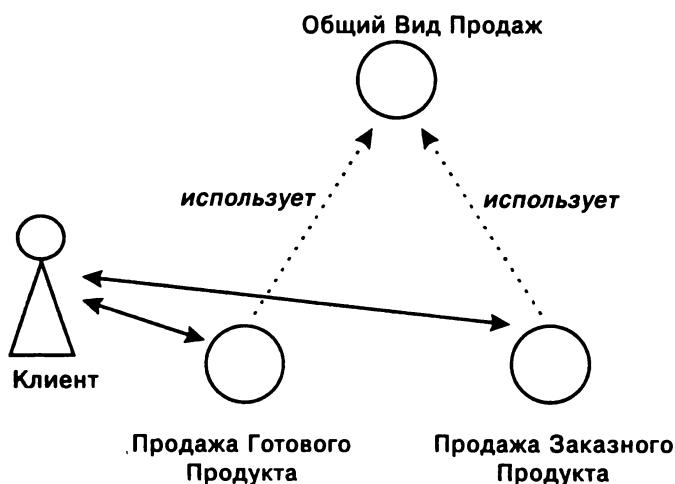


Рис. 7.2. Пример отношения Использование для структурирования прецедента Продажа

много общих частей, например, процедуры оформления самой покупки. Чтобы не описывать общие части дважды, вводят абстрактный класс **Общий Вид Продаж**. Достоинство этого подхода в том, что в будущем изменения придется вносить только в одном месте, а именно в абстрактном классе. Недостаток заключается в том, что клиент должен заранее знать, какой прецедент использовать.

При использовании отношения расширения интерпретация будет несколько другой (рис. 7.3). В данном случае поток событий начинается в прецеденте **Продажа**, то есть клиент не должен заранее знать, с

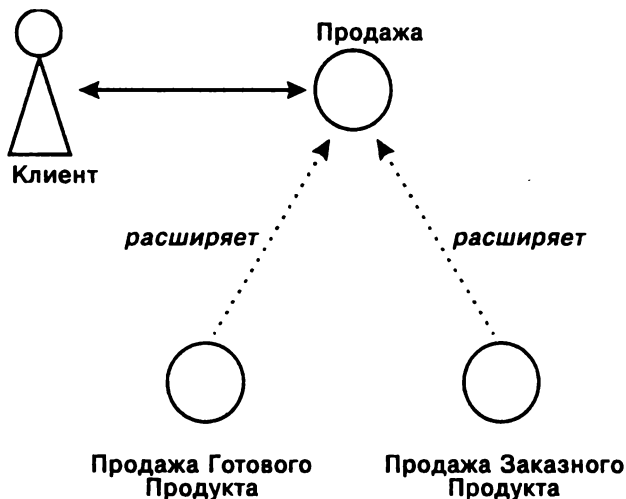


Рис. 7.3. Пример отношения Расширение для структурирования прецедента **Продажа**

каким прецедентом - **Продажа Готового Продукта** или **Продажа Заказного Продукта** - надо работать. Первое действие прецедента **Продажа** - определение типа клиента. Возможно, что очевидного способа выяснить, следует ли обращаться с конкретным клиентом, как обычно, или требуется более сложная процедура продаж, нет, но как только эта информация получена, в прецедент **Продажа** встраивается тело того прецедента, который требуется. Встраиваемый прецедент содержит выполнение полного потока событий. Возврат в исходный прецедент **Продажа** происходит только после завершения всех операций встраиваемого. Заметьте, в этом варианте не предполагается, что различные варианты “продажи” имеют общие фрагменты (по этой причине прецедент называется не “Общий Вид Продаж”, а “Продажа”).

Если разработчик считает, что этот тип структурирования не надо показывать на уровне модели, а достаточно иметь структурированный прецедент **Продажа**, то это можно сделать двумя способами.

Можно либо определить один основной поток событий и несколько альтернативных, связанных с ним, либо описать два совершенно раздельных основных потока. Если большая часть потоков событий одинакова, то следует применять отношение использования; если же потоки существенно отличаются или два отдельных потока использовать разумнее, то используют отношение расширения.

После того как выявлены и структурированы “хорошие” прецеденты и команда по реинжинирингу, руководство и другие заинтересованные стороны приняли П-модель, целесообразно распределить обязанности по описанию отдельных прецедентов. Удобно поручить описание прецедента не группе, а одному человеку. Затем полученное описание согласовывается с владельцем процесса.

Для облегчения понимания старайтесь описывать прецеденты достаточно простым языком. Не формализуйте описания слишком сильно. На этом этапе могут найти хорошее применение результаты опросов, проведенных во время разработки схемы компании. Вполне может потребоваться повторный опрос работников, выполняющих старый процесс, чтобы сделать полнее и четче понимание нового. Эти беседы должны быть неформальными, при этом следует показывать, что мнение интервьюируемого для вас очень важно. Цель бесед - понять, как новый процесс может работать практично и эффективно.

Прецеденты должны представлять завершенные потоки событий в бизнесе, от начального до конечного состояний. Работа над описанием каждого прецедента начинается с описания основного или альтернативных потоков событий. После этого идентифицируются и описываются все исключения или условные потоки событий. Важно, чтобы за описание конкретного прецедента отвечал только один участник команды. Затем полноправный владелец, то есть ответственный за бизнес-процесс, проверяет по описанию соответствие каждого прецедента новым целям компании.

7.1.3

Документирование и проверка П-модели

Документация П-модели содержит общее описание всей модели и описание каждого прецедента. Общее описание содержит краткие описания всех субъектов, прецедентов и отношений между ними. Оно должно давать представление о модели в целом. Если нужно получить более подробную информацию по конкретному прецеденту, то необходимо обратиться к его описанию. Для улучшения понимания модели полезно рассмотреть диаграмму, графически иллюстрирующую отношения между субъектами и прецедентами. Такие диаграммы создаются в качестве приложений к общему описанию модели.

Описание прецедента должно содержать следующую информацию:

- список отношений с субъектами;
- список отношений с другими прецедентами;
- подробное описание основных и, возможно, альтернативных потоков событий прецедента;
- подробное описание всех исключений и условных потоков событий прецедента.

Работа по проверке (обсуждению) делится на два этапа: проверка модели и проверка отдельных прецедентов. Проверка П-модели означает контроль правильности выявленных субъектов и прецедентов и того, насколько разумно модель структурирована. На этом этапе надо согласовать краткие описания субъектов и прецедентов в общем описании модели. Проверка обычно выполняется командой по реинжинирингу вместе с исполнительным руководством.

Проверка конкретного прецедента производится тогда, когда ответственный за его разработку решает, что описание его устраивает. В этом случае он собирает совещание. Количественный и качественный состав участников совещания зависит от природы как конкретного прецедента, так и бизнеса в целом, но обычно на нем должны присутствовать представители следующих групп:

- команда по реинжинирингу;
- персонал, который используется как ресурс при реализации процесса и таким образом напрямую воздействует на то, как выполняются потоки событий прецедента;
- ответственные за другие прецеденты, зависящие от работы рассматриваемого прецедента;
- разработчики информационной системы поддержки.

Описания прецедента должны быть своевременно розданы всем участникам, чтобы они успели их изучить и как следует подготовиться к встрече. До начала обсуждения следует рассмотреть следующие вопросы:

- Корректно ли описано взаимодействие с субъектом?
- На правильном ли уровне детализации описан поток событий?
- Корректно ли и полно описан ход событий?
- Возможно ли упростить и ускорить ход событий?
- Все ли альтернативные потоки событий описаны?

7.2

О-модели нового бизнеса

Существуют два типа объектных моделей, строящихся для описания того, *как* бизнес будет реализовывать прецеденты: идеальная объектная модель и реальная объектная модель. Эти модели имеют несколько разное назначение, но и реальная, и идеальная модели яв-

ляются объектными моделями в том смысле, в каком объектные модели описаны в гл. 3 и 5.

Обе модели используют три типа объектов: интерфейсные объекты, управляющие объекты и объекты-сущности. При работе над идеальной моделью за основу берут прецеденты, а при разработке реальной модели за основу берется идеальная модель. Таким образом, процедуры определения объектов в этих моделях различны.

Рассмотрим различия между идеальной и реальной объектными моделями, затем обсудим их достоинства и недостатки.

7.2.1

Различия между идеальной и реальной объектными моделями

Цель П-модели бизнеса - дать описание того, *что* новый бизнес должен предложить окружению. Это внешний вид компании, так как он описывает ее работу с точки зрения окружения. Объектные модели описывают, *как* новый бизнес будет реализовывать прецеденты. Идеальная объектная модель содержит только те объекты, которые необходимы для выполнения прецедентов. Идеальные объекты описывают, как бизнес должен действовать наилучшим возможным образом. В реальной объектной модели рассматриваются ограничения, наложенные на бизнес. Это может означать, что определенные задачи не могут быть выполнены наиболее эффективным способом, как это описано в идеальной объектной модели, а должны быть приспособлены к ограничениям. В реальной модели принимаются во внимание следующие соображения, не рассматриваемые в идеальной:

- Является ли бизнес географически распределенным или централизованным. К распределенному бизнесу требования обычно строже. К расширению бизнеса надо готовиться всеми возможными способами. Однако нельзя полностью учесть затраты на распределение бизнеса до того, как в нем возникнет нужда.

- В идеальной модели предполагается, что работники обладают определенной компетенцией и в состоянии выполнять поставленные задачи. В реальной компании компетенция служащих может быть недостаточной и, возможно, слишком дорого быстро добиться нужного уровня компетенции, так как это может потребовать массового обучения большей части персонала компании, что иногда невозможно из-за денежных и временных затрат.

- В идеальной модели предполагается реализация бизнеса на новой программно-технической базе. Однако реальной компании может потребоваться несколько лет частично использовать устаревшие ИТ. Замена всей компьютерной поддержки сразу может оказаться слишком дорогостоящей (см. гл. 8).

• Возможно, что компания не хочет в настоящее время менять существующий бизнес. К сожалению, есть много политических факторов, способных отодвинуть срок полного внедрения проекта по реинжинирингу.

В реальной модели эти ограничения могут проявляться следующим образом: добавятся объекты, которые не были нужны в идеальной модели; несколько объектов объединятся в один; объект разделится на несколько подобъектов; некоторые объекты заменятся новым; изменятся описания некоторых объектов.

Идеальную модель обычно гораздо легче понять, чем реальную. Последняя сложнее, так как в ней моделируются ограничения бизнеса. Не всегда очевидно, почему существуют те или иные ограничения и почему реальная модель выглядит именно так. Другая причина того, что реальная модель сложнее, - это больший уровень ее детализации. Это обуславливается тем, что внедряться будет именно реальная модель.

Идеальная модель показывает направление, а реальная - насколько далеко в этом направлении продвинулись. Если решено, что идеальная модель изменяться не будет, то не следует забывать о целях компании, в противном случае результаты будут не совсем теми, какие планировались.

Команда по реинжинирингу должна решить, какие модели создавать, и обсудить достоинства и недостатки различных способов моделирования бизнеса. Команда должна установить, следует ли разрабатывать:

- только идеальную объектную модель;
- только реальную объектную модель;
- идеальную и реальную объектные модели.

Только идеальная объектная модель. В большинстве проектов по реинжинирингу эта альтернатива рассматривается как наиболее подходящая. Команда по реинжинирингу получает четкую картину того, как задачи связаны с продуктами и поставками, не вдаваясь в лишние детали. Идеальная модель описывает простым и не слишком подробным образом объекты, которые должны составить П-модель бизнеса.

Чтобы воплотить новую структуру в бизнес, необходимо создать описания рабочих потоков (см. подразд. 7.2.5), подробно показывающих то, как работа будет выполнена на практике. Перед тем как это сделать, необходимо рассмотреть влияние реализации идеальной модели на компанию. При этом пропускают этап составления реальной модели и сразу переходят к описаниям рабочих потоков. Полученные таким образом описания обеспечивают общение новой компании с основными работниками. Достоинства этой модели в том, что с ней легко общаться, ее просто модернизировать, и время создания такой модели невелико.

Недостаток разработки только идеальной модели состоит в том, что модель довольно абстрактно отражает отношение к действительному состоянию бизнеса.

Только реальная объектная модель. Разница между этим и предыдущим вариантом заключается в том, что реальная модель описывает, как влияют на реализацию модели ограничения в бизнесе. В этом случае начинают с определения идеальных объектов, а затем, по мере введения ограничений бизнеса последовательно заменяют их реальными. Идеальную объектную модель рассматривают как чисто рабочий вариант, который затем трансформируется в реальную модель, показывающую, как все будет работать на практике.

Недостаток создания реальной модели заключается в том, что она требует больше времени для создания, а реальная ее ценность не бесспорна. Стоит заметить, что создание только реальной модели не освобождает от необходимости описания рабочих потоков (см. подразд. 7.2.5), которые должны быть сообщены сотрудникам и поняты ими.

Идеальная и реальная объектные модели. В тех случаях, когда принято решение строить обе объектные модели, создаются последовательности версий модели бизнеса. В первой версии идеальная модель рассматривается как отправная точка, которая преобразуется в реальную модель. Идеальная модель рассматривается как желаемая цель, показывающая направление, в котором следует продвигать бизнес. Если в дальнейшем цели изменяются, то идеальная модель используется как основа для совершения нужных изменений до их включения в реальную модель. Чтобы оправдать наличие более одной модели бизнеса, дополнительная ценность должна превышать затраты на создание второй модели.

Главная проблема при работе с двумя моделями - копирование при модификации. Это трудно делать, работая даже с одной моделью, и более чем в два раза сложнее, если моделей две. Приходится иметь дело и с отношениями между моделями. Изменения в одной из моделей могут потребовать изменений в другой; для идеальной модели последнее утверждение верно всегда; для реальной это не так. Кроме того, требуется обеспечивать поддержку и обновление зависимости между моделями.

Обычно сначала строят идеальную объектную модель, хотя команда по реинжинирингу имеет, разумеется, полное право выбора лучшего варианта в каждом конкретном случае. Вне зависимости от комбинации выбранных моделей необходимо хорошо адаптировать к бизнесу разработанную информационную систему поддержки (ИСП).

Далее в этой главе описывается построение идеальной модели, и вопросы, касающиеся построения реальной объектной модели, более не рассматриваются.

Выявление и описание объектов

Выявление “хороших” объектов (см. для сравнения 7.1.2.) - это искусство, которым непросто овладеть сразу. Ниже описывается метод, который дает хорошие результаты даже для тех, кто не владеет объектно-ориентированным мышлением.

Как правило, выявление ряда объектов-сущностей является очевидным. Примеры таких “очевидных” объектов - это продукция, документы и действия, которые должны быть выполнены как часть повседневной работы или по требованию закона или властей.

Когда получен первый список очевидных объектов-сущностей, следует проанализировать прецеденты. Просмотрите описания всех прецедентов и выявите объекты, которые ответственны за ту или иную часть потока событий прецедента. Для каждой пары субъект/прецедент можно назначить интерфейсный объект. Этот объект обслуживает все взаимодействия с субъектом и по возможности большую часть внутреннего потока событий. Если нецелесообразно или невозможно, чтобы интерфейсный объект выполнял внутренний поток событий, то назначается управляющий объект (или несколько).

Таким образом, описание потока событий прецедента делится на следующие части:

- Описание поведения, напрямую связанного с задачами, требующими контакта с субъектами. За этот тип поведения отвечают интерфейсные объекты.
- Хорошо определенные задачи, которые лежат внутри бизнеса и не требуют прямого контакта с субъектом, например работа специалиста. За этот тип поведения отвечают управляющие объекты.
- Сущности в бизнесе, которые обрабатываются прецедентом. Это могут быть продукты, документы, контракты и т.д. Они становятся объектами-сущностями.

Ориентированный на процессы способ мышления имеет целью добиться того, чтобы один ресурс мог выполнить как можно большую часть процесса. Следовательно, в идеале вся работа, связанная с прецедентом, должна выполняться интерфейсным объектом (или если этот прецедент общается с несколькими субъектами, то несколькими интерфейсными объектами). Если задачи в прецеденте разделены на несколько объектов, то есть риск, что при реализации прецедента потребуются разные виды ресурсов. Это будет означать передачу работы между ресурсами (исполнителями), что увеличит время выполнения прецедента.

В тех случаях, когда нет возможности поручить задачи, связанные с прецедентом, одному ресурсу или нескольким однотипным ресурсам (если прецедент очень сложен или некоторые задачи требуют

компетентных специалистов), следует ввести ресурс, отвечающий за клиентов. Задачи, включающие общение с клиентами, будут моделироваться интерфейсным объектом. Прочие задачи, внутренние по отношению к бизнесу, моделируются управляющими объектами. Интерфейсный объект в этом случае выполняет роль буфера между клиентом и остальной частью бизнеса. Чтобы иметь возможность отвечать на вопросы клиента и помогать в решении его проблем, ресурс, реализующий интерфейсный объект, потребует доступа к ресурсам, реализующим управляющие объекты и, возможно, к информационной поддержке, которая использует эти объекты. М.Хаммер и Дж.Чампи [2] называют этот тип интерфейсного ресурса “менеджер прецедента”.

Рассмотрим, как можно с помощью объектов описать два варианта процесса Продаж: Продажа Готового Продукта и Продажа Заказного Продукта. В обоих случаях сразу видно, что для поддержки контакта с клиентом требуется интерфейсный объект Продавец. При определении объектов прецедента Продажа Готового Продукта нет причины разделять работу на несколько ресурсов. Интерфейсный объект Продавец проверяет наличие продукта и получение заказов клиентами. Если продукцию надо послать по почте, то заказ на поставку можно рассматривать как отдельную рабочую задачу. В этом случае появится дополнительный интерфейсный объект, Отправитель Продукта (рис. 7.4). В приведенном примере продукт моделируется как объект-сущность. Этот объект нужен для хранения информации о продукте, такой, как модель, цвет, цена и комплектация.

При выявлении объектов прецедента Продажа Заказного Продукта результат получится более сложным (рис. 7.5). Действительно, работа по выполнению такого процесса продаж сложнее, и нецелесообразно воз-

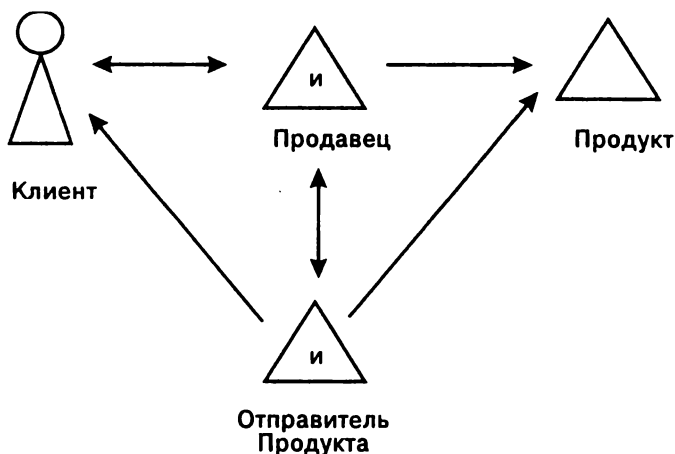


Рис. 7.4. Объекты прецедента Продажа Готового Продукта

лагать всю ответственность на интерфейсный объект Продавца. Следует разделить обязанности между двумя объектами. Теперь Продавец отвечает за контакты с клиентом и прием (обслуживание) заказа, а управляющий объект Проектировщик Продукта занимается подгонкой продукта под требования клиента. При больших размерах заказов необходимо иметь уверенность в том, что продукция будет отправлена клиенту. Следовательно, и в этом примере потребуется интерфейсный объект Отправитель Продукта, который выполняет заказ на поставку продукта клиенту.

Целесообразно ввести в схему объекты-сущности Клиент и Заказ. Первый представляет нужные данные о клиенте, второй о заказе, сделанном клиентом. В прецеденте Продажа Готового Продукта такие объекты вводить не надо, так как там клиент непосредственно получает требуемый продукт, а если его нет, то он ждет, пока продукт появится.

Указать заранее требуемый уровень детализации достаточно трудно. Плохо иметь очень маленькие объекты: модель будет трудна для понимания. Слишком большие объекты не лучше, так как в этом случае высок уровень абстракции. Задачи, моделируемые объектом, должны иметь понятную связь друг с другом. Используйте мнемонические имена объектов, говорящие о задачах, выполняемых этими объектами. Если в один объект собрано несколько несвязанных рабочих задач, будет не совсем понятно, за что же он на самом деле отвечает. Это усложняет обновление модели в целом.

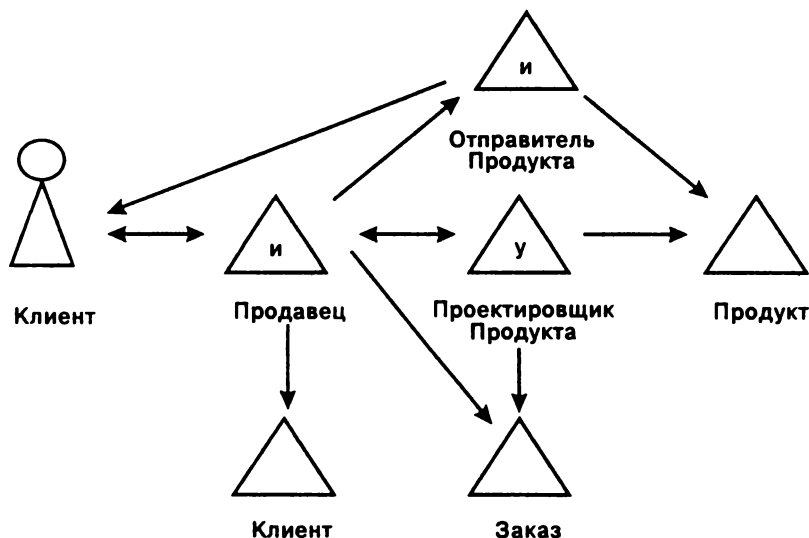


Рис.7.5. Объекты прецедента Продажа Заказного Продукта

Определение отношений между объектами. Динамические отношения между объектами, или отношения коммуникации, выявляются при исследовании прецедентов для определения объектов и при анализе способа взаимодействия этих объектов.

Что касается статических отношений, то определение отношений “ссылка”, “состоит из” и “наследование” производится на следующей стадии, в ходе проверки объектной модели в целом. Отношение “ссылка” необходимо, чтобы показать, что объекты должны что-то знать друг о друге, отношение “состоит из” показывает, что один объект содержит другие, отношение “наследование” указывает использование общих свойств объектов. Определения и иллюстрации этих отношений приведены в гл. 5.

Каждому выявленному объекту дается краткое описание. Прецедент накладывает требования на взаимодействующие объекты, которые должны выполнять его поток событий. Можно считать, что каждый объект, участвующий в прецеденте, играет в нем определенную роль. Каждая такая роль описывается в терминах операций, ситуаций и атрибутов. На этом этапе не следует слишком вдаваться в детали, лучше ограничиться абстрактным уровнем. Важно, чтобы все роли, которые объект должен играть в различных прецедентах, были отражены в его описании.

7.2.3

Моделирование подсистем

Важные действия и обязательства можно легко потерять как при чисто процесс-ориентированной организации, так и при чисто функциональной. Следовательно, даже в модели нового бизнеса надо моделировать и описывать функциональные подразделения компании. Для их представления удобно применять подсистемы, введенные в гл. 5. Подсистемы описывают работы одной компетенции, то есть каждая из них реализуется персоналом с аналогичной подготовкой. Человек, ответственный за какую-либо функцию, отвечает и за совершенствование персонала, занятого в этой функции. Сравните его с владельцем процесса, который отвечает за операции процесса. Они выполняются силами людей, приглашенных из различных функциональных подразделений и, как следствие, обладающих достаточно разным уровнем подготовки.

Процедура разработки и описания функциональной иерархии и того, как процессы в бизнесе работают через функциональные подразделения, выглядит одинаково при моделировании как существующего, так и нового бизнеса (см. подразд. 6.4.1).

Когда в процессе прямого инжиниринга определены и описаны объекты, требуется дополнительная работа по распределению объектов по различным функциональным подразделениям.

Документирование и проверка

О-модель описывается при помощи трех типов документов:

- обзорный документ, перечисляющий все подсистемы и объекты модели, с их кратким описанием;
- описание каждого прецедента, показывающее, как участвующие объекты его выполняют;
- описание каждого объекта, отражающее его роли.

Для большей наглядности целесообразно построить диаграмму, изображающую объекты и их отношения. Очень удобны диаграммы, рассматривающие прецедент с точки зрения участвующих в нем объектов. Эти диаграммы, называемые “Взгляд с точки зрения участвующих объектов”, иллюстрируют, как объекты взаимодействуют при реализации потоков событий прецедентов. Эти диаграммы используются как при идентификации объектов на основе отдельных прецедентов, так и для понимания того, как каждый отдельный прецедент связан с объектной моделью. Более подробная информация о диаграммах такого вида приведена в гл. 5.

Удобно иметь полную картину того, как объекты в модели зависят друг от друга. Если О-модель очень велика, то диаграмму можно разделить на несколько, по одной на каждую подсистему. Диаграммы такого типа называют “Глобальный взгляд на объекты”.

В некоторых случаях, например если объект очень сложен, полезно построить диаграмму, показывающую отношения этого объекта со всеми другими объектами. Такой тип диаграмм очень четко показывает объекты, от которых данный объект зависит, и объекты, которые зависят от него. Этот тип диаграмм называется “Локальный взгляд на объекты”.

После документирования О-модели осуществляется проверка (обсуждение) модели. Эта работа делится на три части: обсуждение О-модели в целом, обсуждение объектов, участвующих в прецеденте, и обсуждение каждого объекта. Во всех трех случаях обсуждение, как правило, проводят люди из команды по реинжинирингу. Объектные модели не обсуждают со всеми сотрудниками, так как единственные, кто действительно имеет к этому отношение и достаточно компетентен, - это команда по реинжинирингу. Однако в команде должны присутствовать люди, которых персонал компании знает и которым доверяет. Эти люди должны понимать основы объектно-ориентированного подхода.

Обсуждение О-модели в целом включает проверку основательности модели, то есть того, что объекты определены правильно (в соответствии с данными выше рекомендациями), что нет различных объектов, отвечающих за одно и то же, что все объекты имеют уникальные имена и что структуры наследования и иерархии включения корректны.

Проверка объектов, участвующих в прецеденте, включает проверку реализуемости хода событий прецедента выбранными объектами и проверку того, что требуемые поведение и обязательства (ответственность) распределены и описаны между этими объектами. Проверяющий должен иметь доступ к описанию прецедента взаимодействующих объектов и к описаниям участвующих объектов. Цель проверяющего - убедиться, что участвующие объекты обеспечивают правильное выполнение всех фрагментов потока событий прецедента, включая альтернативные и необязательные потоки, и обеспечить рассмотрение взаимодействующих объектов вместе с описанием прецедента, включение требуемых объектов и отношений между ними. После этого проверяется, содержат ли описания участвующих объектов все требуемые обязательства.

Рассмотрение отдельных объектов включает проверку того, что описания объектов согласованы и однородны. Различные прецеденты, в которых объект участвует, накладывают на него разные требования, и эти требования ни в коем случае не должны конфликтовать.

7.2.5

Описание потоков работ

Когда требуется донести информацию о новом бизнесе до персонала, объектных моделей недостаточно. Обычно описание объектов дополняют описанием потоков работ, которые специально адаптированы для этой цели. Объектная модель используется в первую очередь для адекватного представления работы бизнеса с точки зрения команды по реинжинирингу, затем на ее основе создаются описания потоков работ, предназначенных для сотрудников. Эти диаграммы бывают различного вида [4].

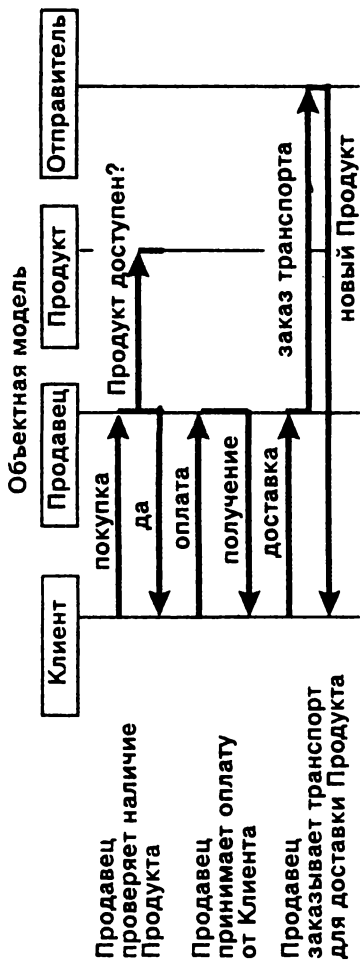
Нет ничего необычного в том, что существуют совершенно разные описания бизнеса для тех, кто занимается моделированием, и тех, кто использует результат. Аналогичная ситуация существует при разработке программных систем, где используется документация на программу и руководство пользователя. Объектная модель соответствует документации, понятной разработчику, а описание потоков работ - руководству пользователя, понятному персоналу (пользователям).

7.3

Диаграммы взаимодействия

Одна из наиболее важных методик определения ролей и их обязательств, которые играет объект, - построение диаграмм взаимодействия. Диаграммы взаимодействия также очень удобны для понимания взаимодействия между потоком событий прецедента и со-

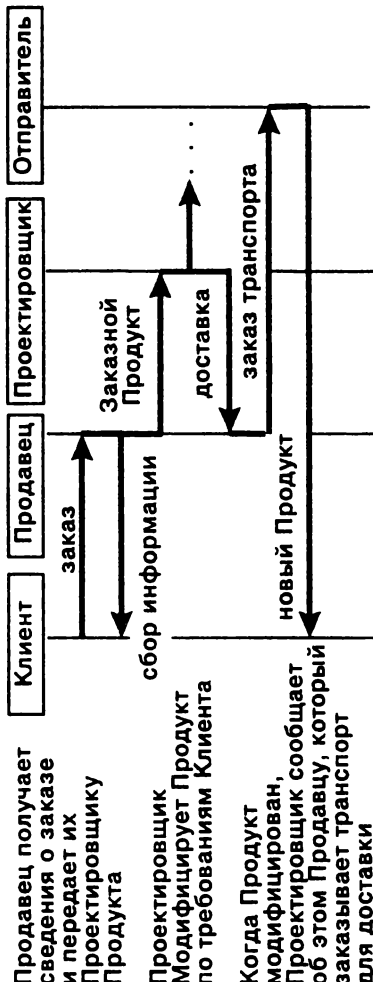
Объектная модель



П - модель

Продажа Готового Продукта

Продавец проверяет наличие требуемого Продукта на складе, затем принимает деньги за него и сообщает Отправителю количество Продукта и адрес Клиента.



Продажа Заказного Продукта

Продавец получает информацию о заказе и передает Проектировщику. Проектировщик модифицирует Продукт в соответствии с требованиями Клиента. Продавец сообщает Отправителю количество Продукта и адрес Клиента.

Рис. 7.6. Поток событий описывается взаимодействующими объектами на одной или нескольких диаграммах взаимодействия

ответствующими объектами. В гл. 5 дано описание диаграмм взаимодействия в технических терминах. В этом разделе будет обсуждено, как и когда можно применять эту методику для распределения обязательств между объектами (рис. 7.6).

В левой части диаграммы находятся описания прецедентов Продажа Готового Продукта и Продажа Заказного Продукта в том виде, как они приведены в П-модели. Эти описания не претендуют на полноту, они просто демонстрируют, как прецедент накладывает требования на участвующие в его выполнении объекты. В правой части диаграммы можно увидеть, как описания этих прецедентов выглядят в объектной модели.

Так как прецедент часто включает несколько альтернативных потоков событий, то обычно из практических соображений описания разбивают на несколько диаграмм взаимодействия. Каждая диаграмма получает уникальное имя для каждого прецедента. В одной диаграмме взаимодействия описывается основной поток событий, а все альтернативные потоки для облегчения понимания - в отдельных диаграммах.

Только после того, как нарисованы диаграммы взаимодействия для потоков событий всех прецедентов, можно быть уверенным в том, что все роли, которые играет данный объект в системе, и, следовательно, его обязательства (ответственность) выявлены. На рис. 7.7 показано, как требования диаграммы взаимодействия (рис. 7.6) влияют на объект Продавец.

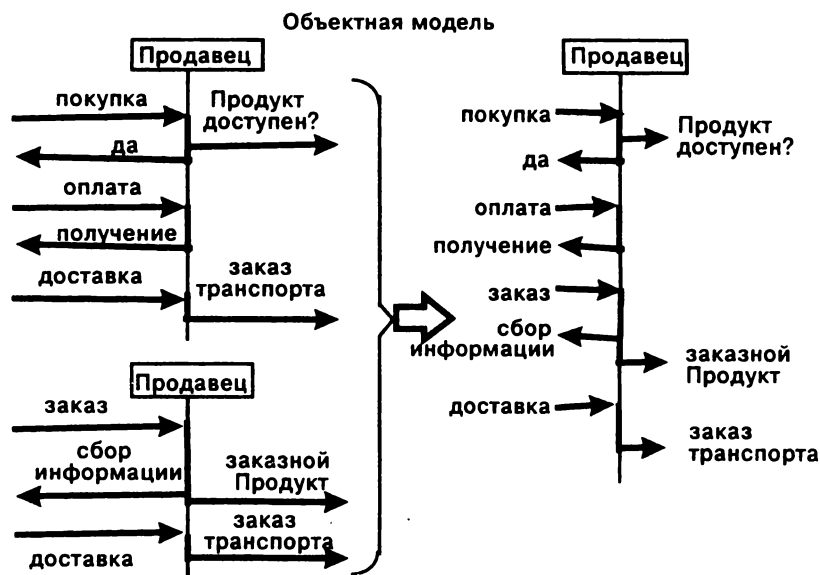


Рис.7.7. Обязательства объекта, определяемые из диаграмм, где он фигурирует

При распределении ответственности между людьми, осуществляющими разработку прецедента, и специалистами, которые отвечают за разработку объектов, участвующих в этом прецеденте, могут возникнуть трудности. Диаграмма взаимодействия - это часть документации по прецеденту, поэтому ее должен строить тот, кто отвечает за прецедент. Когда диаграмма взаимодействия создана, поток событий прецедента распределяется между участвующими объектами. Это означает, что человек, ответственный за прецедент, накладывает требования и обязательства на соответствующие объекты. Все подобные требования к объекту собираются в документ, называемый спецификацией объекта. Затем человек, отвечающий за объекты, формулирует эти требования в описании объекта.

При использовании описанной выше методики есть риск увлечься деталями: на самом деле совершенно не обязательно слишком глубоко вникать в описание ответственности объекта. Определение всех операций, представляющих ответственность объекта, и параметров этих операций обычно заводит слишком далеко. Конечно, иногда детализация описания нужна, но делать ее без веских причин не стоит.

7.4

Разработка информационной системы

Одна из работ по созданию новой компании - это разработка информационной системы поддержки. Новый бизнес не может быть внедрен до ее создания. Разработка информационной системы и создание моделей на начальной фазе являются итеративными процессами с сильной взаимозависимостью. Далее они проводятся в основном параллельно. Так как объектная модель компании и модель информационной системы поддержки сильно зависят друг от друга, их конечные варианты должны быть получены почти одновременно.

Таким образом, очень важно, чтобы модели компании и информационной системы позволяли легко общаться двум группам людей - команде по реинжинирингу бизнеса и разработчикам информационной системы. Для этого в обоих видах деятельности предлагается использовать одинаковые (или почти одинаковые) языки моделирования (см. гл. 8).

7.5

Проверка нового бизнеса

Созданные модели должны быть хорошо проверены на раннем этапе. Если реконструируемый бизнес вместе с интегрированной системой поддержки регулярно не проверять, то в принципе невозможно

но предсказать, как он будет работать. Следует подчеркнуть, что необходимо несколько проверок. После каждого тестирования следует анализировать результаты и модифицировать модель бизнеса.

Т.Давенпорт [1] описывает методику тестирования, которую он называет *“организационное прототипирование”*. Он определяет прототип как “маломасштабную квазиоперационную версию нового процесса, которая может быть использована для тестирования различных аспектов проекта”. По Т.Давенпорту, существует несколько уровней прототипирования: компьютерное моделирование (моделирование новых процессов на раннем этапе с использованием компьютера), тестирование процессов на бумаге (информационный тест процессов при помощи ручки и бумаги), изолированный прототип процесса (создание изолированных прототипов процессов), прототип с интерфейсом (прототипы процессов с интерфейсом к другим процессам и ИСП) и полномасштабный прототип (финальный интегрированный тест). При этом на разных стадиях моделирования применяются разные тесты. Первые тесты помогают отследить ошибки и просчеты на раннем этапе, но они менее точны, чем последующие. Проверка нового бизнеса осуществляется на следующих этапах разработки:

- моделирование бизнеса в идеальной объектной модели, когда еще не осуществлено распределение работ между людьми, машинами или ИСП;
- создание прототипа ИСП;
- выполнение пробной инсталляции модели всего бизнеса, но в изолированной части компании.

Моделирование идеальной модели. Как только разработаны первые варианты идеальной модели, начинается моделирование новых процессов. Моделирование производится до того, как осуществлено распределение работ, образующих процессы, между людьми, ИСП или оборудованием.

Моделирование помогает удостовериться в том, что различные шаги процесса целесообразны. Присваивая этим шагам различные значения, можно измерить параметры рассматриваемого бизнеса и получить оценку его работы.

Инструментальные средства (ИС) моделирования есть на рынке, и во многих случаях может оказаться дешевле купить такое средство, чем пытаться создать свою программу моделирования. Используя интегрированные ИС (как, например, ReThink - см. разд. 9.3), можно быстро получить желаемый результат.

При отсутствии эффективного и хорошо адаптированного средства моделирования следует выполнять процесс вручную, при помощи ручки и бумаги. Эта старомодная “проверка на столе” иногда оказывается достаточно эффективной, хотя она не дает ответов на многие вопросы.

Создание прототипа ИСП. Разработчики ИСП должны определить те части процессов, которые будут реализованы с помощью ИСП. После определения требований к ИСП строится П-модель и создается набросок интерфейса пользователя. После обсуждения создается прототип.

Цель создания такого прототипа - дать пользователям ИСП возможность почувствовать, как ИСП будет работать. В этом случае пользователи уже на раннем этапе смогут предлагать свои варианты улучшений. Кроме того, прототип ИСП позволит измерить, насколько новая ИСП реально влияет на ваш бизнес.

Используя прототипы ИСП, можно экспериментировать, выясняя, увеличивается ли эффективность процесса при расширении тех его частей, которые управляются ИСП.

Выполнение пробной инсталляции. Как только информационная система построена и проверена, выполняются интеграционные тесты с участием реальных пользователей. Пробная инсталляция процессов осуществляется для отдельного небольшого бизнеса и моделирует взаимодействие с клиентами и поставщиками. Сначала по очереди проверяются отдельные процессы, затем тестируется несколько процессов, не имеющих взаимодействий. После этого тестируются взаимодействующие процессы. Результаты этих тестов позволяют выявить различные проблемы и ограничения, для устранения которых потребуется вернуться к моделям и к работе по построению информационной системы. После нескольких итераций компания будет готова к внедрению нового бизнеса во всей организации.

ЛИТЕРАТУРА

1. Davenport T. H. **Business Innovation, Reengineering Work through Information Technology.** - Boston, MA: Harvard Business School Press, 1993.
2. Hammer M. and Champy J. **Reengineering the Corporation: A Manifesto for Business Revolution.** - N-Y: HarperCollins, 1993.
3. Harrington J. **Business Process Improvement.** - N-Y: McGraw-Hill, 1993.
4. Johansson H., McHugh P., Pendlebury J. and Wheeler III W. **Business Process Reengineering. Breakpoint Strategies for Market Dominance.** - Chichester: John Wiley & Sons, 1993.
5. Rockhart J. F. and Short J. E. **Information Technology and the New Organization: Towards More Effective Management of Interdependence//Working Paper CISR 180, MIT Sloan School of Management, Center for Information Systems Research.** - Boston, 1988.

8

ПОСТРОЕНИЕ ИНФОРМАЦИОННОЙ СИСТЕМЫ ПОДДЕРЖКИ

8.1

Модели бизнес-процессов и информационные системы

Информационная технология является основой реинжиниринга бизнес-процессов. Информационная поддержка решающим образом влияет на функционирование процессов и при правильном использовании приводит к многократному повышению их результативности. Учитывая широкомасштабные преобразования, которым подвергается бизнес в ходе реинжиниринга, необходимо как можно раньше определить, какой будет информационная система поддержки. Это позволит уже на ранних этапах учитывать влияние ее на реконструируемые процессы. Для краткости будем ИСП называть “информационной системой”, или “системой”.

Следует отметить, что использование информационной технологии еще не является достаточным условием для успешного проведения реинжиниринга. Для этого необходима и мощная технология разработки информационных систем, обеспечивающая поддержку большинства (если не всех) потребностей компании, включая методики создания распределенных систем - от простых “клиент-сервер” приложений до сложных географически распределенных систем. Создаваемые на основе этой технологии информационные системы должны быть гибкими и легко модифицируемыми, позволяющими отслеживать непрерывные изменения в бизнесе. В настоящее время общепризнанно, что при построении информационных систем необходимо использовать объектно-ориентированную технологию.

Однако объектно-ориентированная технология - это не волшебная палочка, которая позволит решить все проблемы. Она представляет собой лишь основу (см. разд. 3.5 и 3.6), которая должна быть

дополнена и расширена другими технологиями и методиками таким образом, чтобы адекватно представлять всю сложность реального бизнеса (см. разд. 9.2 и 9.3). В этой главе описана техника объектно-ориентированного инжиниринга [7].

Методики, которые можно использовать в объектно-ориентированном инжиниринге бизнеса, подобны методикам объектно-ориентированной разработки программного обеспечения. Действительно, оба процесса хотя и различаются по своему назначению, имеют общую основу - объектно-ориентированные методики построения сложных систем. Эти методики конструируются так, что они обеспечивают поддержку работ по реинжинирингу бизнеса. Они тесно взаимосвязаны, поскольку характер бизнес-процессов определяет требования к информационной системе поддержки, а сама система, как было показано выше, влияет на функционирование этих процессов. В разработке программного обеспечения объектно-ориентированные методики применяются довольно широко, поскольку они уже давно продемонстрировали свое преимущество по сравнению с другими известными методиками. Использование одних и тех же подходов для построения моделей как бизнес-системы, так и соответствующей информационной системы значительно упрощает взаимосвязи между этими моделями.

Для представления зависимостей между моделями будем использовать отношения взаимосвязей (traceability relations), устанавливаемые между явлениями разных моделей. С одной стороны, этот подход облегчает формулировку требований к программному обеспечению, которые возникают в ходе реинжиниринга. С другой стороны, он позволяет выявить влияние новой информационной системы на деятельность компании с тем, чтобы определить направления модификации исходных бизнес-процессов. Подобная итерация между моделью бизнес-системы и моделью информационной системы должна проводиться на каждом шаге разработки, так как только в этом случае удастся согласовывать модели на соответствующем уровне детализации. Чем дальше продвигается разработка, тем более детализованными становятся обе модели и тем труднее их модифицировать.

8.2

Разработка программного обеспечения

Разработка программного обеспечения - это бизнес-процесс в компании, занимающейся созданием программных продуктов. Следовательно, этот процесс может быть представлен прецедентом в модели бизнеса такой компании. Субъектами такого прецедента бизнес-системы являются клиенты компании. Входом прецедента

“Разработка программного обеспечения” являются требования к программному продукту, а выходом - новая версия этого продукта.

Построение информационной системы - это процесс создания моделей, описывающих различные стороны этой системы. Подобно бизнес-системе, разработка информационной системы тоже имеет своих участников (владелец процесса, участник процесса, лидер процесса, аналитик, проектировщик и др.), каждый из которых может быть или не быть ее субъектом. Участник - это некто, выдвигающий требования к информационной системе или просто по тем или иным причинам заинтересованный в ее освоении. Различные участники имеют дело с разными аспектами системы, поэтому они нуждаются в различных ее моделях.

Для каждого типа участников необходимо создавать собственные специализированные модели. Рассмотрим участников разработки бизнес-системы и поддерживающей информационной системы. В бизнес-системе всех участников можно разделить на субъектов (например, таких, как клиенты) и ресурсы (ответственные за процессы, торговые агенты и др.). В случае разработки информационной системы типичными участниками являются пользователи, которые заказывают ее, владельцы процессов, разработчики и др. Некоторые участники бизнес-системы одновременно являются пользователями информационной системы. Поэтому существуют большие преимущества в использовании одних и тех же методик для построения моделей обеих систем.

Ниже будут подробно описаны участники, для которых создается информационная система и ее модели, затем рассмотрены шаги, выполняемые в ходе построения этих моделей, и сами модели. Далее, в разд. 8.3 представлена простейшая объектная бизнес-модель разработки программного обеспечения.

8.2.1

Участники и используемые ими модели

Наиболее важные участники разработки информационной системы поддержки - заказчики и пользователи. В контексте реинжиниринга бизнес-процессов заказчики являются одновременно владельцами бизнес-процессов, а пользователи информационной системы - участниками этих процессов. Как правило, их не интересует, как будет реализована информационная система. Для них самое важное - то, что система сможет делать и как ее использовать. Для них необходимо разработать различные сценарии использования системы. Эти сценарии должны быть представлены в терминах П-модели. Субъекты в таких моделях отражают разнообразные роли, которые могут играть участники бизнес-процессов и пользователи информационной системы.

Другой тип участников - это системные аналитики. Им необходима модель, на основе которой они смогут создавать гибкие, легко-модифицируемые системы. Эта модель не должна иметь дело с деталями реализации. Важно выделить независимые функциональные группы в системе, такие, чтобы внесение изменений в приложение не воздействовало бы на всю систему в целом. Для того чтобы удовлетворить этому требованию, целесообразно разрабатывать идеальные объектные модели информационной системы, т.е. модели, абстрагирующиеся от особенностей конкретной реализации.

Проектировщикам в отличие от аналитиков нужны модели, позволяющие описывать различные способы выполнения системы вплоть до создания программного кода. Объектно-ориентированные языки позволяют получить только описание классов, но не информацию о взаимодействии соответствующих классов. Следовательно, проектировщикам необходимо знать, каким образом классы взаимодействуют для реализации прецедентов. Они нуждаются в реальной объектной модели, представляющей собой некоторую абстракцию программного кода. Реальная объектная модель необходима также специалистам, которые тестируют готовые программы. В этой модели принимаются во внимание такие важные технические аспекты, как среда разработки (язык реализации, доступные вычислительные ресурсы, используемая СУБД и т.д.) и ее ограничения. Результатом проектирования приложения являются исходные тексты программ, которые образуют модель реализации системы.

Члены команды по тестированию системы нуждаются в модели, на основе которой они могли бы провести подробные испытания и убедиться в соответствии программного продукта исходным требованиям. Основу для спецификации тестовых задач представляет собой П-модель информационной системы. Каждому возможному экземпляру прецедента будет соответствовать одна тестовая задача. В совокупности спецификации тестов образуют "оттестированную модель системы".

8.2.2

Этапы разработки моделей ИСП

Каждая из рассмотренных выше моделей разрабатывается и поддерживается совокупностью объектов бизнес-системы. Их задача состоит в реализации программного обеспечения модели бизнеса рассматриваемой компании. Вследствие сложности системы эти объекты обычно объединяются в подсистемы, каждая из которых соответствует отдельной модели. Таким образом, выделяют следующие подсистемы: Сбор Требований, Анализ, Идеальное Проектирование, Реальное Проектирование, Реализация и Тестирование (рис. 8.1).

Объектная модель бизнес-системы, которая будет описана в разд. 8.3, является упрощенной моделью разработки ПО. Она не включает указанные подсистемы, но вводит все объекты непосредственно.

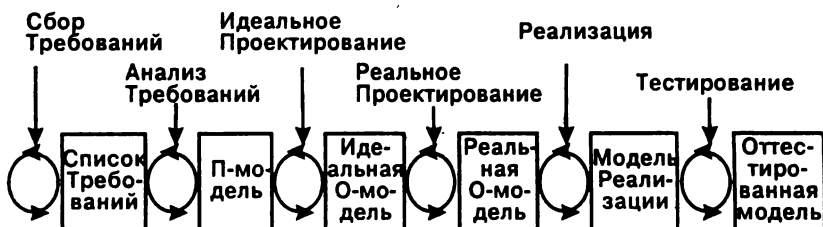


Рис.8.1. Основные этапы разработки программного обеспечения

Описание прецедентов создаваемой информационной системы определяется типом системы, используемыми языком программирования и методиками, а также размерами самой компании. При этом совершенно не обязательно, что будут разработаны все модели, перечисленные в данном разделе.

Рассмотрим этапы разработки, выполняемые при создании моделей информационной системы.

① **Сбор требований.** На этом этапе собираются все предложения, идеи, пожелания и требования ваших клиентов. Список рекомендаций, собранный в результате опросов, составляет промежуточный результат работы. Далее этот список прорабатывается, и для наиболее важных его компонентов составляются спецификации, которые образуют окончательный список требований к информационной системе.

② **Анализ требований.** Цель этого этапа состоит в проверке полноты и непротиворечивости спецификации требований. Далее строится П-модель информационной системы. На этом этапе может быть разработан прототип графического интерфейса пользователя. Результатом этапа является П-модель информационной системы, включающая прецеденты, субъекты и описание интерфейсов конечного пользователя.

③ **Идеальное проектирование.** Предназначено для построения гибкой структуры информационной системы, обеспечивающей простоту ее модификации. Результатом является идеальная объектная модель системы.

④ **Реальное проектирование.** Здесь разрабатывается модель, служащая базисом для реализации информационной системы. Результат - реальная объектная модель информационной системы.

⑤ **Реализация.** На этом этапе создается программный код информационной системы.

6 Тестирование. Проверяется соответствие информационной системы предъявляемым к ней требованиям. Результатом этого этапа должны быть корректные результаты прогона системы на тестовых задачах.

Выше намеренно даны одни и те же имена понятиям моделирования бизнеса и разработки программного обеспечения, которые соответствуют друг другу. Благодаря этому будет легко разобраться в типах объектов и отношений для моделей информационных систем по аналогии с моделями бизнес-систем. Чтобы отличать объекты моделей информационной системы от объектов бизнес-системы, будем в тексте давать явные указания об их принадлежности.

8.2.3

Жизненный цикл разработки программного обеспечения

Все информационные системы изменяются в течение своего жизненного цикла. Для многих систем изменения оказываются наиболее дорогостоящими составляющими жизненного цикла. Тем не менее большинство известных методов разработки программного обеспечения ориентируются на создание новых систем и уделяют недостаточное внимание вопросам их модификации и развития.

Как правило, в начале работы над проектом требования к информационной системе полностью сформулировать не удастся. По мере продвижения работы знания о системе увеличиваются. В результате программный продукт может устареть уже к моменту своего появления на свет: когда первая версия информационной системы создана и используется, появляются новые требования и изменяются старые.

В некоторых случаях предпочтительнее шаг за шагом разрабатывать информационную систему, охватывающую сначала наиболее значимые прецеденты, а затем в ходе практического использования системы и накопления определенного опыта добавлять новые прецеденты (см. разд. 8.5).

Итак, разработка информационной системы обычно сводится к созданию очередной ее версии. С этой точки зрения новая разработка оказывается частным случаем первой версии. Следует отметить, что первая версия имеет важную особенность по сравнению с последующими: именно при выполнении первой версии принимаются основополагающие решения относительно архитектуры информационной системы, которые будут поддерживаться в течение всего ее жизненного цикла.

Если перейти к терминологии моделирования бизнеса, то можно сказать, что заказчик прецедента “Разработка программного обеспечения” является владельцем процесса в данном бизнесе. Новая вер-

сия информационной системы поставляется бета-пользователю, которым является тщательно выбранный участник процесса в бизнесе.

Новая версия информационной системы является результатом выполнения очередного шага “Разработка программного обеспечения”. Если изменяется или добавляется только одно требование к системе, то пересмотр всех моделей может не понадобиться. Например, в случае низкой производительности системы можно искать причину медленной работы с помощью развернутого анализа всех имеющихся моделей, но скорее всего все дело в неэффективном коде или неудачных пользовательских интерфейсах. В первом случае достаточно пересмотреть реальную объектную модель и модель реализации, в то время как во втором анализ и модификация затронут все модели информационной системы.

8.2.4

П-модель информационной системы

В П-модели субъекты представляют роли, которые могут исполнять пользователи системы, а прецеденты - те действия, которые они могли бы производить над ней. Каждый прецедент - это завершенный поток событий в информационной системе, рассматриваемый с точки зрения пользователя. Для представления субъектов и прецедентов будем использовать обозначения, введенные в гл.5 (см. рис. 5.2).

Прецеденты в информационной системе играют две важные роли:

① Прецеденты естественным образом описывают функциональные требования к ИСП. П-модель определяет систему через множество прецедентов. Окружение этой системы определяется описанием различных субъектов, которые используют эту систему так, как это описано в прецедентах. Поскольку П-модель не заменяет О-модели, П-модель должна быть дополнена одной или несколькими объектными моделями. Таким образом, П-модель описывает внешний взгляд на информационную систему, а объектные модели - взгляд изнутри.

② Прецеденты структурируют каждую объектную модель. Если в случае “игрушечных” систем это не обязательно, то для реальных задач без структурирования не обойтись. Существуют сотни примеров объектных моделей для таких задач, как планирование морских путешествий, управление конференциями или системой домашнего отопления, но эти приложения не обладают сложностью реальных систем, применяемых в банковском деле, страховании, обороне или телекоммуникациях. К сожалению, методы, хорошо работающие в простых приложениях, могут оказаться непригодными в более сложных случаях.

Чтобы справляться со сложностью реальной информационной системы, полезно структурировать ее О-модели таким образом, чтобы каждая из них представляла некоторый отдельный аспект ее использования. В описываемом подходе каждый аспект соответствует одному прецеденту информационной системы, и его описание включает только те объекты, которые участвуют в этом прецеденте. Разумеется, один объект может использоваться разными прецедентами. Полная О-модель системы, обладающей реальной степенью сложности, рассматривается с точки зрения разных объектных моделей - одной для каждого прецедента. Для каждого объекта можно полностью выявить его обязательства (ответственность) (см. например, [9]), просматривая все прецеденты, в выполнении которых он участвует. Роль объекта в каждом отдельном прецеденте означает некоторое его обязательство, а совокупность всех ролей характеризует его интегрированные (полные) обязательства.

П-модель информационной системы управляет работой с другими моделями. Именно она направляет работу со всеми другими моделями: структурирует идеальную объектную модель, реализуется в реальной модели, выполняется путем создания программного кода и, наконец, служит основой для построения тестовой документации и инструкций по использованию информационной системы. П-модель описывает все, что ИСП способна делать с точки зрения конечного пользователя.

8.2.5

Идеальная объектная модель информационной системы

В этой модели используются объекты трех типов: интерфейсные, управляющие и объекты-сущности (см. рис. 5.3).

Интерфейсные объекты (ИО) поддерживают взаимодействие информационной системы с ее *окружением*, обеспечивая преобразование и трансляцию данных и событий из внутрисистемных форм представления во внешние и наоборот. Эти объекты позволяют явно выделить зависимую от окружения составляющую модели информационной системы, в то время как управляющие объекты и объекты-сущности не зависят от окружения. Обычно к интерфейсным объектам относят экраны (окна), протоколы связи, интерфейсы с датчиками и устройствами вывода на печать.

Объекты-сущности (ОС) представляют малоизменяемые сущности ИСП, “живущие” дольше, чем экземпляры прецедентов, в которых они участвуют. Они используются для представления атрибутов, отношений и поведения тех или иных явлений, событий и персонала. Важно отметить, что их поведение нередко оказывается столь же сложным, как и поведение других объектов в системе. Часто объекты-сущности представляют предметы, участвующие одновременно в

нескольких прецедентах. Значения их атрибутов обычно вводятся пользователем в ходе диалога с прецедентом информационной системы. Объекты-сущности могут понадобиться и для решения внутренних задач системы. Если некоторые объекты-сущности бизнес-системы должны обрабатываться информационной системой, они должны быть включены в модель ИСП в виде объектов-сущностей.

Управляющие объекты (УО) используются для описания поведения системы, задаваемого прецедентами. Обычно они управляют другими объектами и выполняют роль координаторов в модели. УО описывают в отличие от ИО и ОС не зависящее от окружения поведение, т.е. поведение, не зависящее от того, каким образом осуществляется обмен данными и результатами между системой и ее окружением: электрическими сигналами, процессорными прерываниями, вводом с клавиатуры, вызовом процедур или выбором из меню.

Причина, по которой введены три типа объектов, состоит в стремлении обеспечить гибкую структуру модели, допускающую ее модификацию и развитие. В ходе эксплуатации информационной системы неизбежны текущие изменения; известно, что даже стабильные системы время от времени требуют локальных модификаций. Чаще всего проводится модификация пользовательских интерфейсов, а также тех или иных функциональных возможностей информационной системы, при этом изменения затрагивают только интерфейсные объекты ИСП. Изменения потока событий в прецеденте влияет на поведение управляющих объектов в информационной системе, а изменения объектов-сущностей в модели бизнеса отражаются на объектах-сущностях ИСП. Поскольку последнее случается относительно редко, объекты-сущности оказываются самыми стабильными элементами модели информационной системы.

8.2.6

Реальная объектная модель

В реальной объектной модели работают только с реальными объектами. Обычно в первом варианте реальной модели каждый реальный объект соответствует в точности одному объекту идеальной модели. В дальнейшем количество реальных объектов растет за счет добавления новых или расщепления существующих объектов. При реализации реальному объекту будет соответствовать один или несколько классов. Целесообразно один из классов определить как общий (public), а другие - частными (private). Общий класс, помимо других функций, обеспечивает интерфейс с окружающим миром, а с помощью остальных описывается реализация реального объекта. В частных классах можно, например, описать детали работы в распределенной архитектуре или способы периодического сброса теку-

щей информации в реляционную базу данных. Конкретный вид отношения между объектами реальной модели и объектами в реализации, безусловно, зависит от используемого языка программирования. Поэтому прежде чем начинать работу над реальной объектной моделью, следует решить, как это отношение должно выглядеть.

Взаимодействуя друг с другом, реальные объекты обеспечивают выполнение прецедентов информационной системы. При этом каждый реальный объект играет одну или несколько ролей, по крайней мере, в одном из прецедентов. К сожалению, не существует единого метода построения множества “хороших” реальных объектов на основе идеальной модели. Решающую роль здесь играет язык программирования, выбранный для реализации информационной системы.

Что считать “хорошим” реальным объектом, зависит от окружения. Размер реальных объектов, например, определяется стилем программирования на этом языке. То, что считается правильным и хорошим для Ады, может оказаться неверным для Smalltalk'a. Поскольку реальные объекты непосредственно отображаются в конструкции языка программирования, необходимо строить их таким образом, чтобы получить “хороший” код.

Хотя особенности используемого языка влияют на модель реального объекта, важно обеспечить ясность и легкую модифицируемость модели. В любом случае проектируйте объекты с инкапсуляцией (см. разд. 3.5 и [1]), даже если язык программирования не поддерживает подобные конструкции.

Для удобства управления ИСП она разбивается на подсистемы, группирующие ее объекты в отдельные пакеты в соответствии с тем или иным критерием. Подсистемы используются как в идеальной, так и в реальной модели. Иерархическая структура реальной модели может быть рекурсивной: выделенные подсистемы, в свою очередь, могут состоять из других подсистем.

8.2.7

Модель реализации информационной системы

Модель реализации информационной системы представляет собой аннотированный исходный код программ. Любой язык позволяет применить объектно-ориентированные методики моделирования, но объектно-ориентированные языки имеют большие преимущества по сравнению с остальными, облегчая перевод основных конструкций модели в конструкции языка.

Оттестированная модель является завершающей в процессе разработки программного обеспечения. Она описывает спецификации и результаты тестовых прогонов. Таким образом, фундаментальными понятиями здесь являются тестовые спецификации и результаты тестирования. Тест, подобно прецеденту, может рассматриваться как объект. Тестовая спецификация может рассматриваться как класс, и это позволяет применять механизм наследования для формирования сходных тестов, различающихся в небольших деталях.

Объекты бизнес-системы, разрабатывающей программное обеспечение

Идеальные объекты бизнес-системы, участвующие в прецеденте Разработка Программного Обеспечения, изображены на рис. 8.2. Здесь для ясности изложения основной идеи приведены только семь типов объектов. На практике полная О-модель бизнес-системы по разработке программного обеспечения включает сотни типов объектов, из которых около 30% относятся к интерфейсным и управляющим, а все остальные - к сущностям.



Рис.8.2. Интерфейсные и управляющие объекты, участвующие в прецеденте Разработка Программного Обеспечения

“Интервьюер”, занимающийся сбором требований, является абстрактным интерфейсным объектом бизнес-системы. Он наследуется всеми интерфейсными объектами модели бизнес-системы, которые тем или иным образом взаимодействуют с клиентами этой системы. “Ответственный за бета-тестирование” взаимодействует с субъектом “Бета-пользователь”, представляющим ключевых пользователей разрабатываемого продукта, являющихся, по сути, активными участниками разработки этого продукта.

8.3.1

Сбор требований

Цель сбора требований состоит в получении пожеланий, предъявляемых к информационной системе клиентами и самой компанией. Полученные сведения используются для выявления важнейших требований и определения примерных сроков разработки. Результатом работы является список требований.

Наличие модели бизнес-системы в значительной мере помогает сформулировать требования к функциям информационной системы. Кроме этих требований, необходимо собрать и более специфические требования об используемом оборудовании, базах данных и т.д.

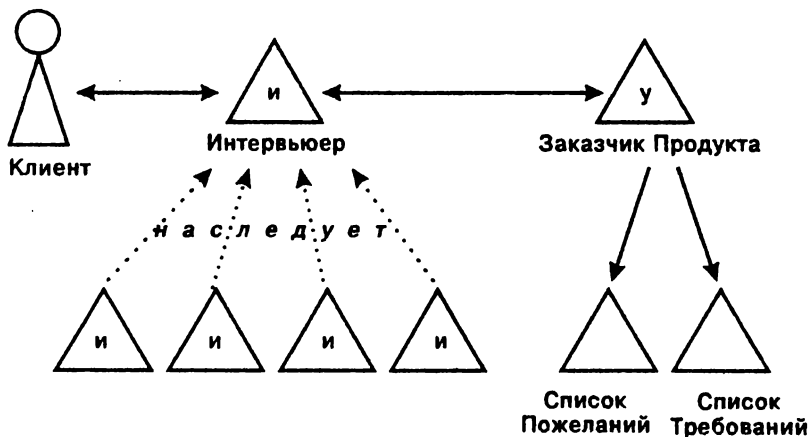


Рис.8.3. Объекты бизнес-системы, участвующие в Сборе Требований

Объекты бизнес-системы, описывающие сбор требований, изображены на рис. 8.3 (стрелки обозначают коммуникации). Интерфейсный объект Интервьюер - абстрактный объект, наследуемый

всеми интерфейсными объектами модели бизнес-системы, которые взаимодействуют с клиентами компании. Следовательно, каждый из таких интерфейсных объектов участвует в Разработке Программного Обеспечения. Вся информация, собранная объектами бизнес-системы, должна быть обработана. Эту функцию выполняет управляющий объект Заказчик Продукта. Все предложения, поступившие от клиентов, хранятся в Списке Пожеланий, исходя из которого Заказчик Продукта строит Список Требований. Элементы последнего упорядочены по приоритетам и содержат приблизительные оценки сроков разработки.

8.3.2

Анализ требований

Задача анализа требований - установить границы информационной системы и определить, какие функции она должна выполнять. П-модель информационной системы следует рассматривать как своеобразный контракт между разработчиком и заказчиками.

Если модель бизнес-системы существует, то разработка П-модели поддерживающей информационной системы осуществляется непосредственно из О-модели бизнеса (см. разд. 8.4). Если такой модели еще нет, то П-модель строится исходя из представлений потенциальных пользователей системы.

П-модель информационной системы состоит из прецедентов, субъектов и описаний интерфейсов конечного пользователя. Нередко работа начинается с составления набросков интерфейсов, выполняемого совместно с пользователями системы. Затем, после уточнения представлений о системе, приступают к созданию прототипов.

Прототип служит удобным средством коммуникации между разработчиком и потенциальным пользователем. Текстовые спецификации никогда не смогут отразить динамику поведения информационной системы в такой степени, как это делают действующие прототипы. Прототипирование отличается от инкрементной разработки тем, что целью является не создание программного продукта, а сбор требований и демонстрация определенных свойств будущей информационной системы.

В анализе требований участвуют следующие объекты бизнес-системы: Заказчик Продукта, Аналитик, Список Требований, Спецификация Требований и П-модель (рис. 8.4). Заказчик Продукта отвечает за инициирование разработки в соответствии со Списком Требований. Анализ требований имеет итеративный характер. Он описывается следующим образом.

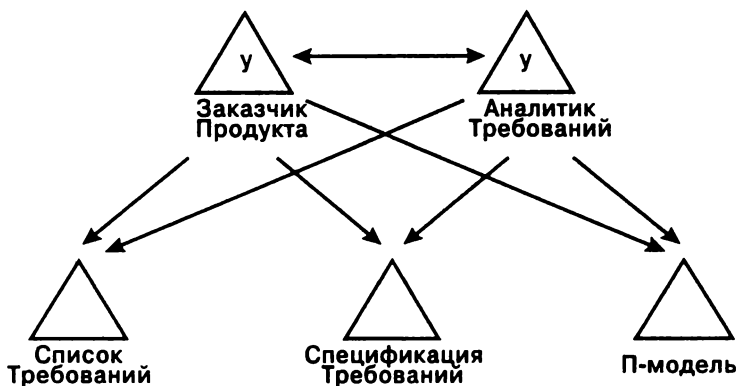


Рис.8.4. Объекты бизнес-системы, участвующие в Анализе Требований

- Заказчик Продукта “инициирует” Аналитика сформировать Спецификацию Требований на основе Списка Требований.
- Аналитик, используя информацию о доступных ресурсах и приоритетах, указанных в Списке Требований, разрабатывает Спецификацию Требований для текущей версии.
- Аналитик начинает с определения потенциальных субъектов информационной системы. Чтобы понять информационную систему, необходимо знать, для кого она создается.
- Аналитик готовит глоссарий терминов, представляющих понятия, используемые при описании прецедентов информационной системы в П-модели.
- Аналитик выделяет прецеденты исходя из потребностей субъектов и Спецификации Требований. При необходимости он структурирует П-модель с помощью отношений использования и расширения (см. разд. 5.5).
- Аналитик подробно описывает прецеденты. Это самый длительный этап в ходе анализа требований.
- Детально описываются интерфейсы конечного пользователя.

Пример (заимствован из [6])

Допустим, что необходимо построить систему управления автоматическими кассовыми аппаратами (АКА) в банке. Клиенты хотят, чтобы система позволяла:

- снимать деньги со счета;
- переводить деньги с одного счета на другой;
- просматривать текущий баланс счета;
- делать вклады.

Разумеется, реальные спецификации составляют десятки страниц, и мы не хотим приводить их полностью. Наша цель - проиллюстрировать основные понятия.

В системе АКА выделим двух субъектов. Первый субъект - Клиент Банка, для которого в первую очередь и создается информационная система. Система АКА должна взаимодействовать с центральной финансовой системой банка, содержащей данные о клиентах и счетах. Эта Финансовая Система есть второй субъект. Выделим четыре прецедента: Снятие Денег, Перевод Денег, Запрос Баланса и Вложение Денег (рис. 8.5 и 8.6).

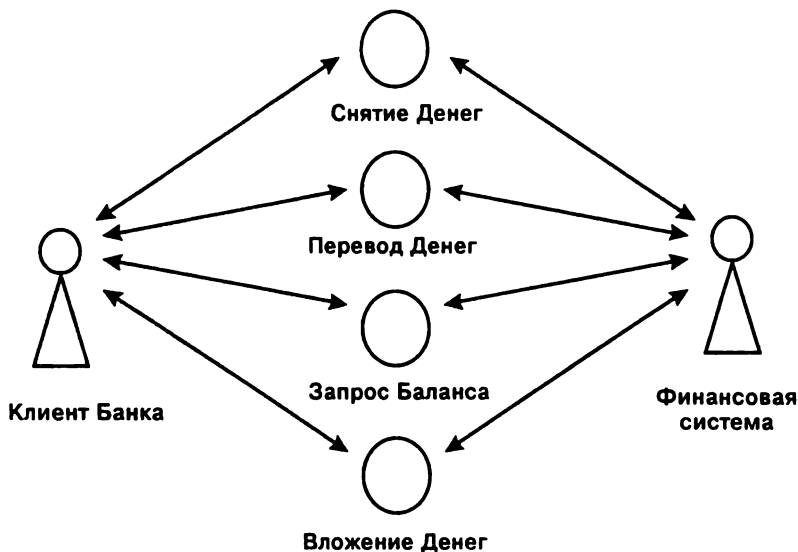


Рис.8.5. П-модель для системы АКА

Описание прецедента Снятие Денег может выглядеть следующим образом.

НАЧАЛО

Снятие Денег начинается с того момента, когда Клиент Банка вставляет кредитную карточку в кассовый аппарат. Система проверяет, имеет ли силу (действительна ли) эта карточка. Если нет, то прецедент завершается (см. ниже КОНЕЦ).

СНЯТИЕ

Система запрашивает информацию о счете и о той сумме денег, которую следует выдать Клиенту Банка. Система повторяет запрос до тех пор, пока сумма не станет кратной \$20. Система проверяет, располагает ли данный кассовый аппарат запрашиваемой суммой денег. Если нет, то прецедент завершается (см. КОНЕЦ). Система АКА обращается к Финансовой Системе с запросом о наличии достаточной суммы денег на счете клиента. Если сумма на счете меньше запрашиваемой, то прецедент завершается (см. КОНЕЦ). Если на счете достаточно денег, то система выдает деньги и печатает расходный ордер. Сумма на счете Клиента Банка уменьшается соответствующим образом.

КОНЕЦ

Аппарат возвращает кредитную карточку; прецедент завершается.

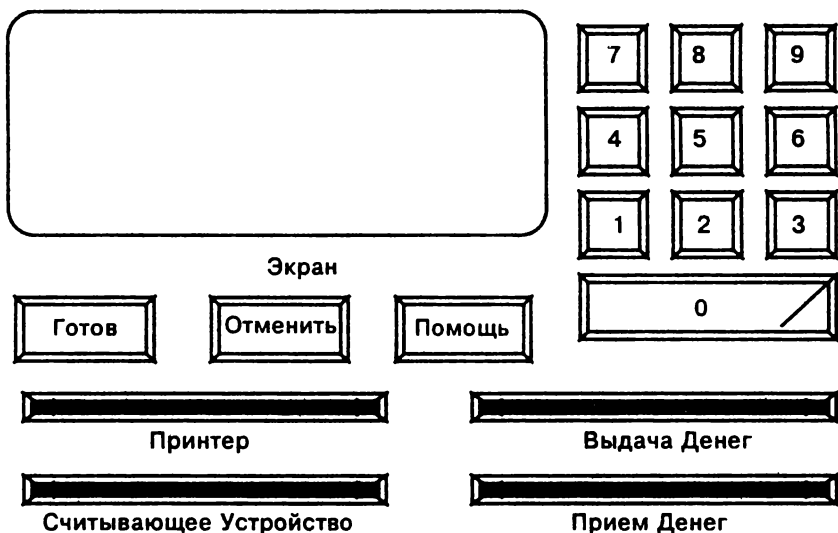


Рис.8.6. Общий вид интерфейса конечного пользователя в АКА

8.3.3

Идеальное проектирование

Основная задача идеального проектирования - определить, как структурировать информационную систему, чтобы обеспечить простоту модификации системы при изменении функциональных требований к ней. В качестве основы для построения идеальной объектной модели будем использовать П-модель. Объекты бизнес-системы, участвующие в идеальном проектировании, изображены на рис. 8.7. Последовательность работ в ходе идеального проектирования можно охарактеризовать следующим образом:

- После того как Аналитик строит первую версию П-модели, инициируется Проектировщик Идеальной Модели. Входом для него являются Спецификации Требований и П-модель.

- Проектировщик Идеальной Модели выделяет объекты-сущности информационной системы, представляющие предметы и явления, явно фигурирующие в П-модели.

- Для каждого прецедента ИСП Проектировщик Идеальной Модели определяет объекты, необходимые для реализации потока событий в нем. Можно начать работы по определению объектов с субъекта, инициирующего прецедент, а затем определить требуемые интерфейсные и управляющие объекты и объекты-сущности.

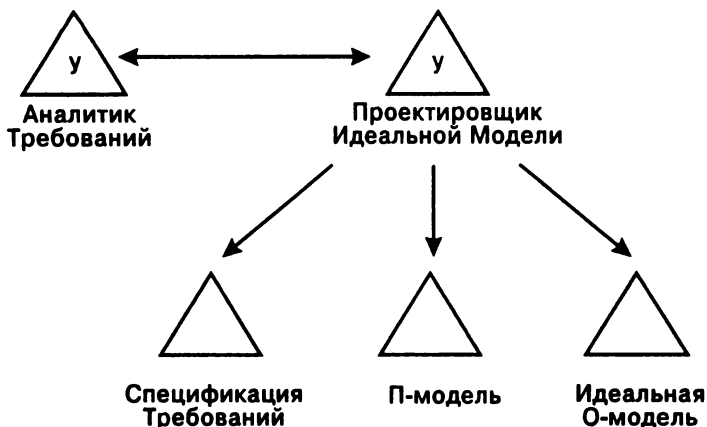


Рис.8.7. Объекты бизнес-системы, участвующие в Идеальном Проектировании

- Проектировщик Идеальной Модели описывает выполнение каждого из прецедентов информационной системы в терминах объектов ИСП.

- Проектировщик Идеальной Модели строит единую модель путем унификации описаний прецедентов. Под унификацией понимают некоторую неформализованную процедуру сборки прозрачной, не избыточной и непротиворечивой модели из отдельных фрагментов, соответствующих прецедентам. Например, несколько сходных объектов, используемых в различных прецедентах, могут быть сведены к одному объекту. Простое правило, которым следует руководствоваться при выполнении унификации, состоит в том, что среди имен объектов модели не должны встречаться синонимы или омонимы.

- Для каждого объекта ИСП отыскиваются все прецеденты, в которых объект выполняет некоторую роль. Проектировщик Идеальной Модели интегрирует эти роли для каждого объекта и на этом основании определяет и описывает обязательства объекта.

Пример

Объекты, участвующие в прецеденте Снятие Денег, приведены на рис. 8.8. С помощью отношений коммуникации изображена схема взаимодействия этих объектов в ходе выполнения прецедента. Приведем описание выполнения прецедента Снятие Денег в терминах объектов информационной системы.

НАЧАЛО

Выполнение прецедента начинается с того момента, когда экземпляр объекта Ввод Кредитной Карточки обнаруживает введенную в аппарат кредитную карточку. Ввод Карточки посылает ее идентификатор и код Обработчику Транзакций, который проверяет, имеет ли она силу. Если карточка

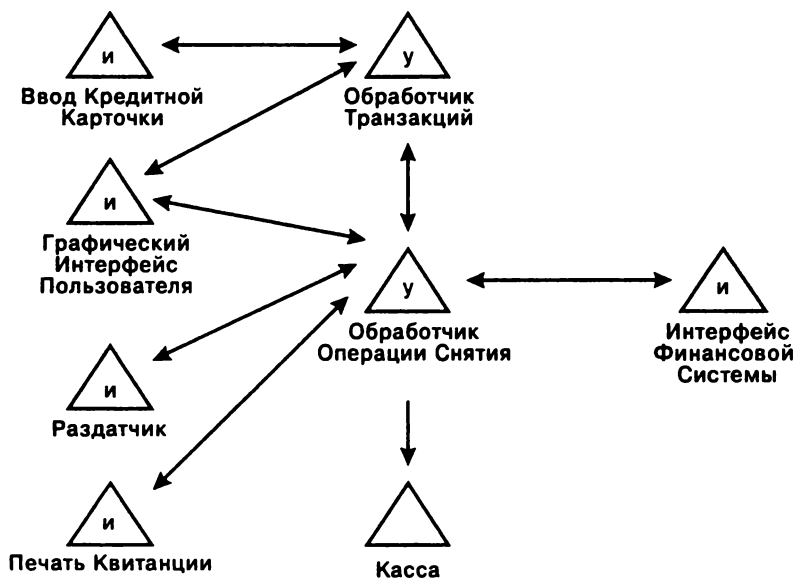


Рис.8.8. Объекты информационной системы, участвующие в прецеденте Снятие Денег

недействительна, то Графическому Интерфейсу Пользователя поручается выдать сообщение об этом клиенту, и сеанс завершается (см. ниже КОНЕЦ).

Если карточка действительна, то Графическому Интерфейсу Пользователя поручается запросить у клиента его персональный идентификатор (PIN). После ввода идентификатора Обработчик Транзакций считывает его и сравнивает с кодом, введенным с карточки. В случае несовпадения клиенту предлагается ввести персональный идентификатор повторно. В том случае, когда в течение трех попыток клиент не может правильно указать свой идентификатор, аппарат оставляет у себя карточку, и сеанс завершается (см. КОНЕЦ).

После ввода правильного идентификатора Обработчик Транзакций поручает Графическому Интерфейсу Пользователя выдать на экран список возможных операций и предлагает клиенту сделать выбор. Ответ клиента возвращается Обработчику Транзакций.

СНЯТИЕ ДЕНЕГ

Если клиент выбрал операцию Снятие Денег, Обработчик Транзакций запускает Обработчик Операции Снятия, который начинает свою работу с проверки наличности в аппарате. Если ее нет, то сеанс завершается (см. КОНЕЦ).

Если в аппарате есть наличные деньги, то Обработчик Операции Снятия запрашивает у Графического Интерфейса Пользователя информацию о номере счета клиента и о требуемой ему сумме. Эта информация должна вводиться клиентом. Запрос повторяется до тех пор, пока требуемая сумма не оказывается кратной \$20.

Если требуемая сумма превышает размер наличности в Кассе, то сеанс завершается (см. КОНЕЦ).

Обработчик Операции Снятия через Интерфейс Финансовой Системы запрашивает подтверждение банка на проведение операции.

Если банк отказывается подтвердить операцию, то Графическому Интерфейсу Пользователя поручается выдать об этом сообщение клиенту, и сеанс завершается (см. КОНЕЦ).

В случае подтверждения Раздатчик выдает деньги, в Кассе изменяется размер текущей наличности, а Печать Квитанции выводит на устройство печати квитанцию для клиента.

КОНЕЦ

Ввод Карточки выталкивает кредитную карточку из аппарата. Выполнение прецедента завершается.

Заметим, что Объект Касса участвует сразу в двух прецедентах информационной системы - Снятие Денег и Вложение Денег. Задача объекта - знать сумму наличных денег, имеющихся в аппарате в текущий момент. Другие объекты информационной системы могут просить кассу добавить или изъять некоторое количество денег или просто сообщить о доступной наличности.

8.3.4

Реальное проектирование

Основная задача реального проектирования состоит в поиске путей эффективной реализации П-модели в условиях существующей среды разработки. Базой реального проектирования являются помимо П-модели идеальная О-модель и спецификация требований (рис. 8.9).

Процесс реального проектирования можно описать следующим образом.

- После того как Идеальная Объектная Модель информационной системы создана, Проектировщик Реальной Модели принимает ее в качестве отправной точки для создания Реальной Объектной Модели. Помимо нее, он использует Спецификацию Требований и П-модель.

- Проектировщик Реальной Модели начинает с адаптации Идеальной О-модели к условиям реального окружения. Например, если выбранный язык программирования не поддерживает механизм наследования, нельзя напрямую использовать этот механизм в реальной модели. В этом случае необходимо найти способ реализовать его с помощью доступных языковых средств. Адаптация подразумевает также решение проблем стыковки информационной системы с СУБД и другим существующим программным обеспечением.

- Для каждого прецедента Проектировщик Реальной Модели описывает с помощью диаграмм взаимодействия поведение взаимодействующих объектов в ходе выполнения прецедента. Эту работу называют проектированием прецедентов.

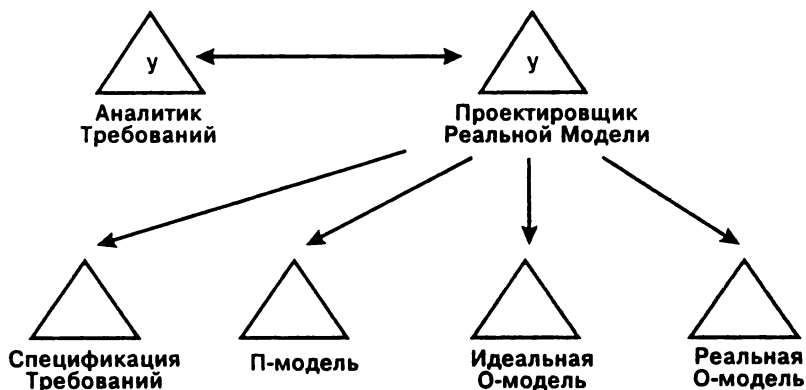


Рис.8.9. Объекты бизнес-системы, участвующие в Реальном Проектировании

- Анализируя диаграмму взаимодействия, Проектировщик Реальной Модели выделяет атрибуты и операции для каждого реального объекта.

Пример

Для описания поведения объектов при выполнении прецедента Снятие Денег составим диаграммы взаимодействия. Для упрощения каждый частный поток событий покажем на отдельной диаграмме (см. рис. 8.10).

Проанализировав, как объект Касса участвует в прецедентах Снятие Денег и Вложение Денег, можно выделить для соответствующего реального объекта следующие атрибуты и операции.

Атрибуты

Имя: Сумма

Тип: Целое

Краткое описание: хранит текущее значение наличных денег в аппарате.

Операции

Имя: Сообщает-сумму

Параметр: Текущая-сумма (ВЫХОД)

Краткое описание: Сообщает текущее количество денег в аппарате

Имя: Добавляет-сумму

Параметр: Приращение-суммы (ВХОД)

Краткое описание: Увеличивает текущее значение наличных денег на величину, указанную на входе.

Имя: Уменьшает-сумму

Параметр: Уменьшение-суммы (ВХОД)

Краткое описание: уменьшает текущее значение наличных денег на величину, указанную на входе.

8.3.5

Реализация

Основой для данного этапа разработки является реальная объектная модель, в которой описаны интерфейсы и поведение реальных объектов (рис. 8.11). Целесообразно установить наиболее простое

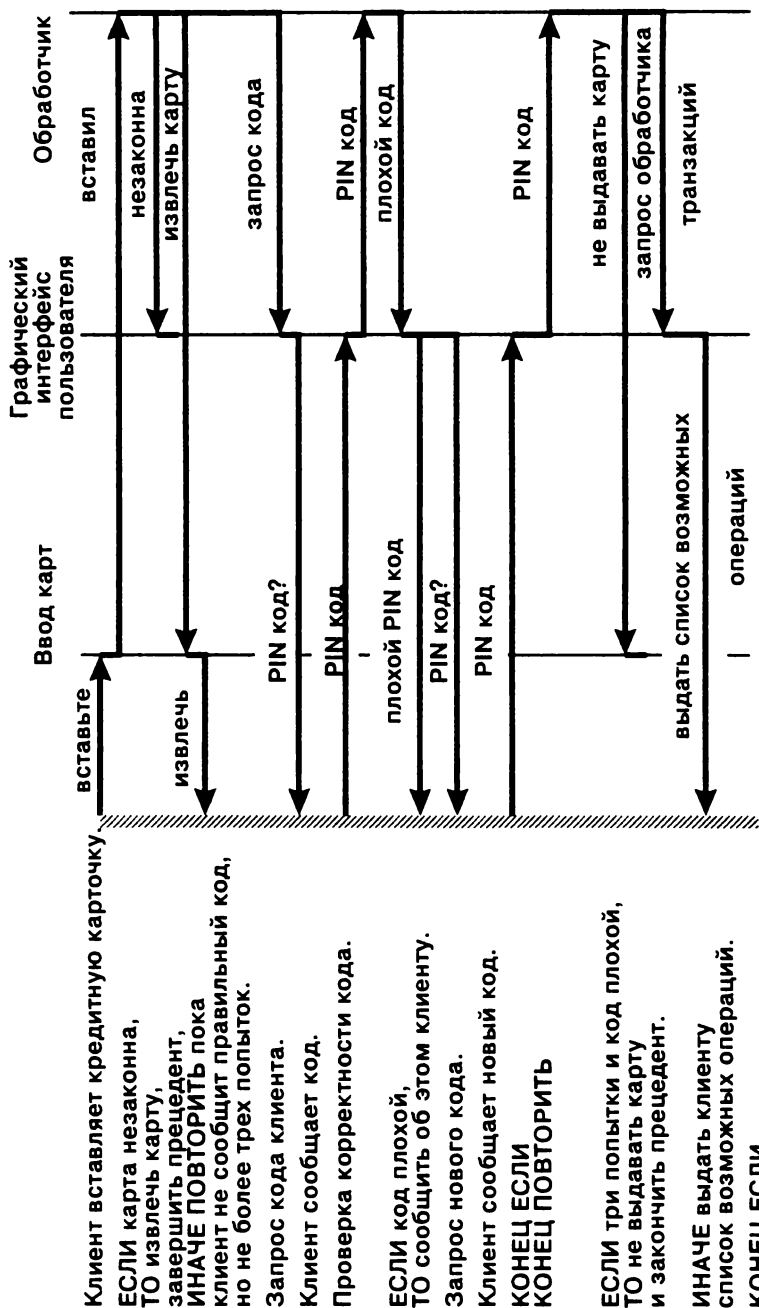


Рис. 8.10. Диаграмма взаимодействия начального потока прецедента Снятия Денег

соответствие между реальным объектом и его реализацией в виде классов. Этого легче всего достичь при использовании объектно-ориентированной среды реализации. К сожалению, на практике это не всегда осуществимо, например, приходится сопрягать О-модель с реляционной БД.

На этапе реализации крайне важна возможность использовать готовые компоненты и расширяемые конструкции (frameworks). **Компоненты** - это завершенные фрагменты, из которых, как из строительных блоков, можно строить информационную систему на более высоком уровне абстракции, чем это допускают языки программирования. Компоненты используются без изменений - в том виде, в котором они были созданы. В отличие от них **расширяемые конструкции** представляют собой подсистемы, спроектированные для повторного использования и расширения. Примерами расширяемых конструкций могут служить библиотеки классов графического интерфейса пользователя, шаблоны объектов в реляционных базах данных, стандартные конструкции для описания распределенных систем или протоколов коммуникации.

При реализации выполняются следующие шаги:

- объекты реальной О-модели преобразуются в программный код;
- для каждого объекта проверяется корректность его внутренней структуры и поведения - тестирование объектов.

Реализация всегда выполняется проектировщиком реальной модели, поэтому в описанной на рис. 8.11 модели бизнес-системы нет объекта типа Разработчик Реализации. Проектировщик Реальной Модели, используя в первую очередь Реальную Объектную Модель, а также Спецификации Требований и П-модель, создает программный код.



Рис.8.11. Объекты бизнес-системы, участвующие в этапе Реализация

Тестирование

Под **тестированием** подразумевается проверка соответствия системы своей спецификации. Этап тестирования требует много времени, и на него обычно приходится 30 - 50% стоимости разработки. Эта цифра, однако, оказывается меньше (около 25%) в случае использования объектно-ориентированных методов создания программного обеспечения. Тестирование - длительный и серьезный этап разработки, который должен быть описан и запланирован так же, как и все остальные этапы.

В большинстве проектов тестирование относится на конец процесса разработки программного обеспечения. Однако сейчас большинство специалистов считает, что для получения кода высокого качества о тестировании следует заботиться с самого начала разработки.

На рис. 8.12 изображены объекты, участвующие в тестировании информационной системы. Тестирующий разрабатывает Тестовые Документы, используя прецеденты из П-модели. И хотя в первую очередь Тестирующий проверяет программный код, он должен иметь



Рис. 8.12. Объекты бизнес-системы, участвующие в Тестировании

доступ ко всем моделям информационной системы, которые были созданы во время ее разработки. Это позволяет ему выявить модель - источник ошибки в программном коде. Во время тестирования Тестирующий должен также взаимодействовать с Аналитиком Требований и Проектировщиком Реальной Модели.

8.3.7

Бета-тестирование

Программный продукт не может считаться готовым до тех пор, пока “настоящий” клиент не признает его работу удовлетворительной. Чтобы избежать неприятных сюрпризов в будущем, необходимо до начала распространения продукта передать его для апробации специально выбранным пользователям. Отличие бета-тестирования от обычного состоит в том, что оно не ограничивается проверкой программного кода, а рассматривает всю систему в целом, включая и документацию. Именно такой способ обеспечения высокого качества продукта называется *Бета-Тестированием*.

Хорошо отработанный процесс разработки должен гарантировать выполнение всех требований, предъявленных к информационной системе. Однако свойственное людям и компьютерам несовершенство требует проверки того, что разработан именно тот программный продукт, который необходим клиенту.

Ответственный за Бета-Тестирование производит бета-версию продукта - Бета-Продукт. Бета-Продукт включает исполняемый код и полный комплект документации. Ответственный за Бета-Тестирование предоставляет Бета-Продукт в распоряжение Бета-Пользователя и обрабатывает все замечания и комментарии пользователя о продукте. При правильном следовании технологии разработки должны быть отработаны все требования и пожелания бета-пользователя. На данном этапе не должны возникать серьезные проблемы.

* * *

Выше описан прецедент Разработка Программного Обеспечения бизнес-системы. В этом описании основное внимание уделено тому, каким образом взаимодействуют друг с другом наиболее важные объекты бизнес-системы, участвующие в процессе разработки (рис. 8.13). Интервьюер собирает все виды пожеланий и требований Клиента. Заказчик Продукта анализирует их, составляет список требований, упорядочивает его по приоритетам и дает приблизительную оценку сроков разработки. Аналитик детализирует детальные спецификации требований к очередной версии продукта и строит на их

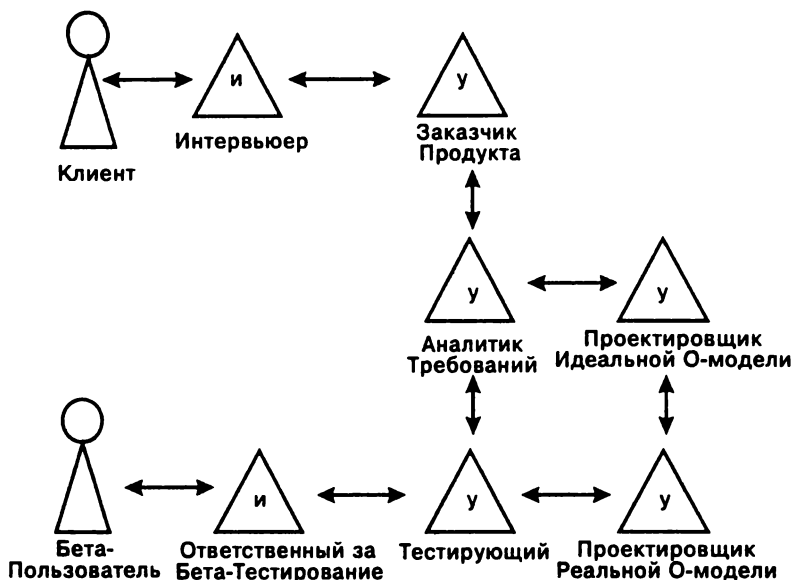


Рис.8.13. Объекты бизнес-системы, участвующие в прецеденте Разработка Программного Обеспечения

основе П-модель. Разработчик Идеальной О-модели создает идеальную структуру системы, на основе которой Разработчик Реальной О-модели проектирует реальную модель. Эта модель, в свою очередь, является базой для реализации. Готовый программный код проверяется Тестирующим, после чего Ответственный за Бета-тестирование передает готовый продукт Бета-Пользователям для окончательной проверки его работоспособности и обработки последних замечаний к системе.

Цель разработки новой версии продукта состоит в удовлетворении всех потребностей клиентов, поэтому очень важно, чтобы клиенты принимали активное участие в разработке и тестировании, а не только в выдвижении требований к системе.

8.4

Разработка информационной системы и разработка бизнеса

Напомним, что такое модель бизнес-системы и что представляют собой ее компоненты. П-модель бизнес-системы представляет взгляд извне на бизнес-процессы компании. Объектные модели дают более

или менее абстрактный взгляд на компанию изнутри. Кроме того, О-модели показывают, как прецеденты реализуются взаимодействующими объектами.

Объект может участвовать в нескольких прецедентах, играя некоторую роль в каждом из них. Полная совокупность обязательств объекта в системе представляется множеством его ролей. Один экземпляр объекта может принимать участие в нескольких экземплярах прецедентов. Таким образом, один прецедент может влиять на другой. Однако подобное влияние может приводить и к конфликтам, которых хотелось бы избежать.

Чтобы избежать от возможных конфликтов, необходимо отслеживать использование ресурсов. Для этого необходимо рассмотреть бизнес-систему на двух уровнях: бизнес-уровень, содержащий модель бизнеса, и уровень ресурсов, описывающий, как ресурсы используются бизнес-моделью.

8.4.1

Реализация модели бизнес-системы с учетом ресурсов

В объектной модели бизнеса используются объекты трех типов: интерфейсные, управляющие и объекты-сущности. Каждому из этих типов объектов соответствуют свои сущности в бизнесе.

Управляющие объекты представляют действия, которые должны выполняться сотрудниками (человеческие ресурсы), использующими те или иные инструменты.

Объекты-сущности представляют предметы и явления в бизнесе: автомобили, документы или нечто более абстрактное, например знания о чем-либо.

Модель бизнес-системы выполняется ресурсами компании. Ресурсы могут быть одушевленными (человеческими) или неодушевленными (компьютеры и информационные системы). Иными словами, модель бизнес-системы включает два уровня: уровень бизнеса и уровень ресурсов. На уровне ресурсов модель описывает, как взаимодействуют между собой сотрудники компаний и как они используют информационные системы для выполнения прецедентов, представленных в П-модели бизнеса. На этом уровне можно обнаружить большое разнообразие типов человеческих ресурсов: сотрудники отдела продаж, проектировщики и разработчики. К неодушевленным ресурсам относятся средства технической поддержки деятельности сотрудников, в первую очередь - информационные системы.

Как соотносятся между собой объекты двух уровней модели? Можно показать, что каждый объект на уровне бизнеса состоит из нескольких объектов уровня ресурсов. Следовательно, между объек-

тами двух уровней имеется отношение “состоит из”. Но использование этого отношения для описания взаимосвязи между объектами разных уровней даст нам немного. Ведь в результате нас интересует не структура объектов уровня бизнеса, а способ их взаимодействия на уровне ресурсов. Поэтому целесообразно отслеживать лишь наличие отношений вне зависимости от их типа: это позволит явно отразить, как объекты бизнес-системы поддерживаются прецедентами и объектами в слое ресурсов информационной системы. Конечно, такой подход не так точен, как в случае использования отношения “состоит из”, но зато он гибче и, следовательно, полезнее.

8.4.2

Алгоритм выделения прецедентов информационной системы из модели бизнеса

Идентифицируя прецеденты информационной системы, исходят из интерфейсных и управляющих объектов О-модели бизнеса. Если уже разработана реальная О-модель бизнеса, то лучше всего использовать именно ее объекты, если нет, можно использовать объекты идеальной модели. Для простоты описания введем понятие “активный объект”, которое будет обозначать интерфейсный и управляющий объекты. Активный объект может рассматриваться таким образом, как суперкласс для интерфейсного и управляющего. Будем также говорить об информационной системе в единственном числе, хотя на самом деле в компании может использоваться несколько таких систем.

Для каждого активного объекта бизнес-системы выполняют следующие действия (рис. 8.14):

- Определяют, реализуется ли активный объект ресурсом, использующим информационную систему.
- Если да, то на уровне ресурсов выделяют соответствующий ему субъект информационной системы. Дают ему то же имя, что и у активного объекта бизнес-системы (на рис. 8.14 Объект X и соответствующий ему Субъект X).
- Если активный объект бизнес-системы использует информационную систему, то для него на уровне ресурсов идентифицируют прецедент информационной системы. В имени прецедента отражают его связь с этим обязательством активного объекта.
- Если активный объект использует информационную систему по-разному в разные моменты времени (на рис. 8.14 Объект X имеет обязательства А, Б, В), то для него идентифицируют несколько прецедентов информационной системы (на рис. 8.14 прецеденты А, Б, В).

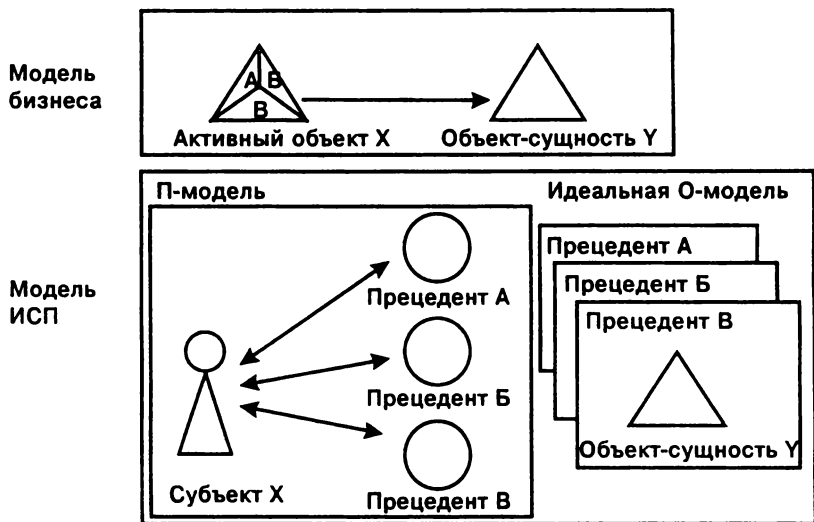


Рис. 8.14. Пример соответствия объектов бизнес-модели субъектам, прецедентам и объектам-сущностям информационной модели

- Просматривают имена, присвоенные прецедентам и субъектам информационной системы на уровне ресурсов. Убеждаются в том, что они достаточно содержательны в контекстах как двухуровневой модели бизнес-системы, так и П-модели информационной системы.

- Повторяют перечисленную выше последовательность действий для всех активных объектов бизнес-уровня.

Объекту-сущности бизнес-уровня обычно соответствует объект-сущность в идеальной О-модели информационной системы. Однако иногда удобно сопоставить атрибутам в модели бизнес-системы объекты-сущности в модели ИСП. Объекты-сущности бизнес-уровня не имеют непосредственного соответствия в П-модели ИСП. К объекту-сущности в модели бизнеса могут иметь доступ различные активные объекты бизнес-системы. Следовательно, соответствующий объект-сущность (или объекты-сущности) в информационной модели может участвовать в нескольких прецедентах ИСП.

Поясним изложенное выше на конкретном примере, в котором в модели бизнеса активный объект X имеет обязательства, обозначенные как А, Б и В. Следовательно, в П-модели информационной системы можно выделить субъект с именем X и прецеденты с именами А, Б и В. Реализуя обязательство В, активный объект X в модели бизнеса обращается к объекту-сущности Y. В идеальной объектной модели информационной системы это соответствует тому, что объект-сущность Y участвует в прецеденте В (см. рис. 8.14).

Предложенный алгоритм не отвечает на вопрос, как следует интерпретировать отношение коммуникации между активными объектами в модели бизнеса? Объекты взаимодействуют друг с другом, и следует выяснить, как это взаимодействие может быть поддержано средствами информационной системы. В частности, если информационная система обеспечивает автоматическую поддержку коммуникации между объектами бизнес-системы, то следует освободить эти объекты от обязательства инициировать коммуникацию. Однако, как правило, это не так, и обычно задача информационной системы состоит в передаче информации между объектами бизнес-системы.

Приведенный выше алгоритм позволяет построить основу модели информационной системы по объектной модели бизнеса. К сожалению, модель бизнеса дает слишком абстрактное представление о том, как сотрудники корпорации намерены использовать информационную систему, и не существует такого алгоритма, который помог бы построить более детальное описание способов взаимодействия пользователей с этой системой.

Построение хороших интерфейсов пользователей ИСП - это целая наука. Моделированию прецедентов должна предшествовать работа, к которой привлекаются пользователи по определению прецедентов ИСП и используемого в них пользовательского интерфейса. Для этого можно организовать серию рабочих встреч, в ходе которых разработчики интерфейсов изучают то, как будущие пользователи выполняют свои обычные обязанности на своих рабочих местах. Разработчики интервьюируют их и просят описать всевозможные сценарии использования создаваемой информационной системы, т.е. описать экземпляры прецедентов информационной системы. Чтобы добиться лучшего взаимопонимания на данном этапе, разработчики представляют пользователям наброски интерфейсов и по достижении более или менее устойчивых решений относительно интерфейсов приступают к созданию прототипов. Более подробные сведения об этом можно найти в [7] и [2].

Пример

В упрощенной бизнес-модели банка выделим двух субъектов: Клиент и Информационное Бюро Банка. Клиент может инициировать один из следующих прецедентов: Получение Кредита (Кредитование), Управление Счетом (Управление) и Инвестирование (рис. 8.15).

- Прецедент Получение Кредита описывает действия, выполняемые при подаче клиентом просьбы о выделении кредита.
- Прецедент Инвестирование показывает, как банк может помочь клиенту выгодно вкладывать его деньги.
- Прецедент Управление Счетом показывает, как банк может помочь клиенту управлять его вкладом, например, помещать деньги или снимать их.

В прецеденте Кредитование присутствуют два субъекта: Клиент и Информационное Бюро Банка. Им соответствуют интерфейсные объекты Обработчик Кредитов (Обслуживающий клерк) и Информационное Бюро Банка. Обработчик Кредитов имеет обязательство по управлению потоком со-

бытий в прецеденте Получение Кредита. Для этого прецедента можно выделить объекты-сущности, обеспечивающие управление Приложением Кредитования; информацию о Кредитном Законодательстве; знания о таких сущностях, как Клиенты и Счета в банке (рис. 8.16).

Обратите внимание, что объект-сущность Клиент отличается от субъекта Клиент. Первый необходим для представления информации обо всех клиентах, зарегистрированных в банке, а второй - для представления клиента, обращающегося в банк для выполнения некоторой операции. Чтобы избежать недоразумений, будем явно указывать, о каком Клиенте идет речь - объекте-сущности или субъекте.

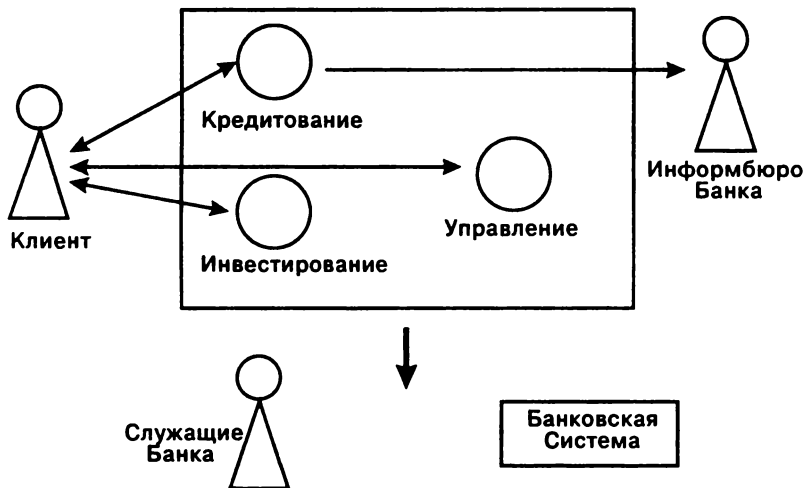


Рис.8.15. Прецеденты бизнес-системы банка, выполняемые Служащими Банка и Банковской Системой

Рассуждая подобным образом, выделяем интерфейсный объект Обработчик Инвестиций для прецедента Инвестирование. Объектами-сущностями здесь являются: Инвестиционный План, Клиент, Счет и Знания-о-Рынке. Можно выделить управляющий объект Аналитик-Рынка для сбора долговременных Знаний-о-Рынке. Для прецедента Управление Счетом можно выделить интерфейсный объект Обработчик Счета и объекты-сущности План Сбережений, Клиент и Счет.

Многие банки предпочитают использовать один и тот же тип ресурса для Обработчика Кредитов, Обработчика Инвестирования и Обработчика Счета. Поэтому в окончательном варианте модели будем работать с одним интерфейсным объектом под названием Обслуживающий Клерк, позволяющим выполнять три различные задания (рис. 8.16).

Рассмотрим теперь вопрос о том, какая требуется ИСП для поддержания описанного выше бизнеса.

Для того чтобы упростить Аналитику Рынка и Обслуживающему Клерку хранение необходимой информации и выполнение возложенных на них обязательств, можно включить в ИСП следующие объекты-сущности: Знания-о-Рынке, Клиент, Счет. Эти объекты-сущности представляют данные,

совместно используемые другими объектами бизнес-системы. Главная задача ИСП - поддерживать выполнение бизнес-процессов, а не только хранить информацию. ИСП должна, например, помогать Аналитику Рынка, используя приобретенные Знания-о-Рынке, выполнять анализ рынка и осуществлять инвестиционные расчеты.

Интерфейсный объект Обслуживающий Клерк принимает участие в трех прецедентах (см. рис. 8.16), т.е. он имеет по крайней мере три обязательства (ответственности). Для того чтобы кредитовать или инвестировать, банк должен проанализировать различные альтернативы, оценить безопасность, риск и т.п. Следовательно, в ИСП целесообразно ввести субъект Обслуживающий Клерк и два прецедента: Приложение кредитования и План инвестиций. Управляющий объект бизнес-системы Аналитик Рынка приводит к введению в ИСП субъекта Аналитик Рынка и прецедента Анализ-Знаний-о-Рынке.



Рис. 8.16. Объекты бизнес-системы, участвующие в прецедентах Кредитование, Инвестирование и Управление

Заметим, что это далеко не полный перечень участвующих объектов. Другие объекты будут обнаружены через прецеденты ИСП. Таким образом, между объектами бизнес-системы и объектами ИСП существует некоторое соответствие.

На рис. 8.17 изображена взаимосвязь между моделями бизнес-системы и информационной системы. С помощью пунктирных линий отслеживается наличие отношений между объектами разных моделей. Интерфейсный объект бизнес-системы Обслуживающий Клерк имеет два обязательства, что

отображено на схеме белым и черным цветами. Каждое из обязательств поддерживается отдельным прецедентом информационной системы. В П-модели информационной системы определен субъект, соответствующий Обслуживающему Клерку. В модели бизнес-системы Обслуживающий Клерк взаимодействует с объектами-сущностями Счет и Клиент. Для обоих имеются соответствующие объекты-сущности в модели информационной системы; оба участвуют в прецедентах информационной системы Приложение по Кредитованию и Разработка Инвестиционного Плана.

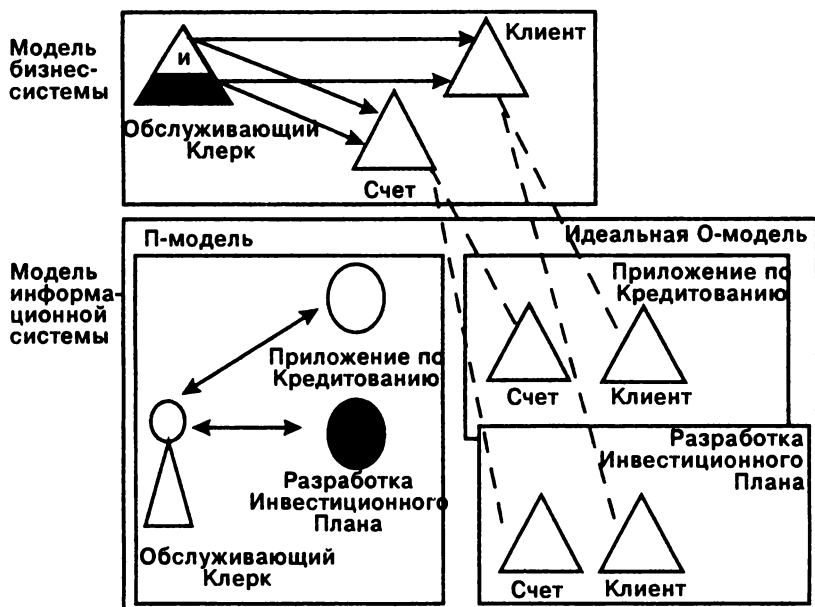


Рис.8.17. Взаимосвязь между моделями бизнес-системы и информационной системы

Для Обслуживающего Клерка необходима следующая поддержка: интерфейсный объект должен контролировать количество денег на счетах клиента, т.е. снимать деньги со счетов, переводить их с одного счета на другой, запрашивать баланс счета, класть деньги на счет, а также запрашивать информацию о состоянии рынка.

Чтобы реализовать все эти функции, необходимо определить дополнительные прецеденты информационной системы: Снятие Денег, Перевод Денег, Запрос Баланса и Вложение Денег (рис. 8.18).

Если сравнить эту П-модель банковской системы с той моделью, которая была разработана для системы АКА (разд. 8.3), то можно обнаружить, что с точки зрения клиента содержащиеся в них прецеденты имеют общее назначение. Но поскольку они различаются по функциям и реализации, в банке применяются две различные информационные системы. Система АКА взаимодействует с Банковской Системой для управления счетами и клиентами, поскольку последняя хранит информацию о них. Таким образом, Бан-

ковская Система оказывается субъектом для АКА и наоборот. Клиент банка может обращаться к АКА непосредственно, в то время как связь с Банковской Системой осуществляется только через клерка.

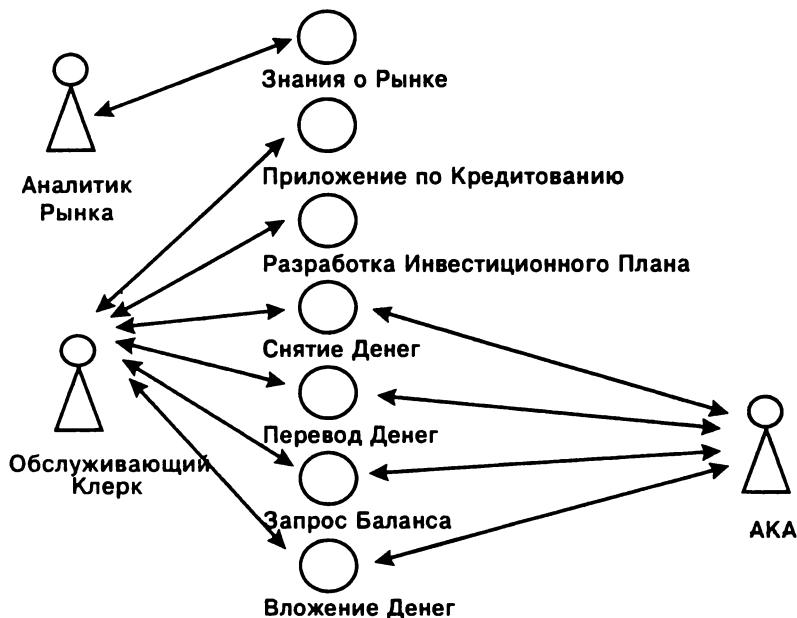


Рис.8.18. Прецеденты информационной системы для банка

8.4.3

Итерации между моделями

Из описанного выше следует, что сначала строится модель бизнес-системы, а затем модели информационной системы. Однако на самом деле это не совсем так. Важно понимать, что разработка моделей осуществляется итеративно и инкрементно. Это означает, что разработка моделей продвигается шаг за шагом и обе модели постоянно согласовываются. Частью этой работы может быть создание прототипов информационной системы, чтобы избежать затрат больших средств на разработку, общий вид которой не определен.

То, что описано в этом разделе, можно назвать последовательностью моделей: П-модель бизнес-системы, Объектная модель (модели) бизнес-системы, П-модель информационной системы, Объектная модель (модели) информационной системы. При этом разработка бизнес-процессов ИСП осуществляется итеративно.

Внедрение новой информационной системы

Предположим, что разработана П-модель и Объектная модель вашего бизнеса. Взяв О-модель бизнеса за основу, разрабатывают П-модель ИСП. После этого появляется ясное представление о требованиях к ИСП. На этом этапе перед компанией встанет вопрос: какой способ получения требуемой ИСП выбрать?

Поскольку каждая компания уникальна, не существует универсальной “хорошей” информационной системы, и не стоит надеяться, что можно купить готовый продукт, отвечающий всем требованиям компании. Скорее всего компании придется разрабатывать эту систему самостоятельно. Но как это лучше сделать? Может быть, в таком случае имеет смысл заказать ИСП и адаптировать ее к новым бизнес-процессам? Следует проанализировать потребности и возможности и выбрать одну из следующих стратегий:

- приобрести готовую поддерживающую информационную систему;
- разработать ИСП самостоятельно;
- провести реинжиниринг существующих информационных систем и модифицировать их.

Если принято решение приобрести готовую систему, то придется адаптировать П-модель ИСП компании к тому, что решено купить. Изменение П-модели ИСП может потребовать изменения бизнес-модели. Это допустимо, если не помешает достижению основных целей бизнеса компании. Адаптация может понадобиться и в том случае, если выбран третий вариант - реинжиниринг и модификация существующей информационной системы.

Среди специалистов по БПР распространено мнение о том, что лучший способ получения хорошей поддерживающей информационной системы состоит в полном отказе от старой системы (существовавшей в компании до проведения реинжиниринга) и установке принципиально новой. На первый взгляд, это решение может показаться самым экономичным, но опыт показывает, что процедура установки и внедрения будет намного стабильнее и безопаснее, если сохранить существующую систему и вносить в нее поэтапные изменения, вместо того чтобы заменить ее сразу и полностью новой неизвестной системой. Даже если реинжиниринг старой системы оказывается в два или в десять раз дороже установки новой, то более важно, чтобы работа компании не прерывалась. Например, возможность продолжения работы со старыми базами данных повышает

надежность и ценность новой информационной системы. Более подробно проблемы реинжиниринга программного информационного обеспечения будут рассмотрены в подразд. 8.5.1.

Другая причина, по которой не стоит отказываться от старой информационной системы, связана с большими временными и материальными затратами, которые потребует ее замена. Каждая компания скорее всего вложила большие средства в создание программного и информационного обеспечения, и вряд ли будет разумно выбрасывать их вместе с устаревшей системой. Таким образом, единственная разумная с практической точки зрения стратегия состоит в пошаговой перестройке существующей системы. Здесь важно иметь в виду, что всегда наибольшие вложения компании делают не в оборудование и программное обеспечение, а в данные. Данные - одна из самых больших ценностей любой компании.

Сама по себе установка информационной системы поддержки еще не приводит к достижению революционных улучшений в бизнес-процессах. Необходимо на опыте определить, каким образом информационная поддержка позволяет изменить текущее состояние дел в лучшую сторону. Как отмечают [4], преимущества информационной технологии заключаются не столько в ускорении работы, сколько в том, что она делает возможным создание нового. Эти же авторы, а также [3] утверждают, что основные новшества в бизнесе появляются только тогда, когда внимательно исследованы информационные технологии и обнаружено то новое, что они могут дать.

Итак, на практике получение новой информационной системы в большинстве случаев сводится к реинжинирингу старой системы. Вряд ли целесообразно пренебрегать затратами на создание старой ИСП без особых причин. При этом следует взять за основу новые модели бизнеса. Если компания действительно хочет добиться новых результатов, то необходимо в первую очередь исследовать возможности информационных технологий в контексте ее бизнеса. Чтобы обеспечить стабильность внедрения новой ИСП, устанавливать ее надо постепенно, шаг за шагом.

Ниже рассмотрены некоторые базовые идеи и принципы, полезные при внедрении информационной поддержки.

8.5.1

Реинжиниринг программного и информационного обеспечения

В рамках описываемого подхода реинжиниринг представляет собой процесс создания абстрактной модели перестраиваемой исходной системы, выявления изменений, которые следует внести в эту модель, и перепроектирования системы [9]. В большинстве случаев

оказывается, что нет смысла заменять старую систему полностью. Лучше всего найти способ постепенной замены старых ее компонентов новыми. Нельзя упускать возможность включения фрагментов существующей системы в новую систему путем видоизменения интерфейсов.

Можно выделить следующие шаги реинжиниринга информационной системы:

- создание приблизительной картины того, что должно быть изменено или добавлено;
- определение прецедентов информационной системы, которые будут затронуты в ходе изменений;
- построение идеальной объектной модели для этих прецедентов;
- выявление взаимосвязей между объектами в идеальной модели и блоками в существующем программном коде;
- определение интерфейсов между старыми и новыми фрагментами системы;
- реализация и подключение измененных фрагментов.

Итак, в первую очередь надо определить хотя бы приблизительно, что следует изменить в старой информационной системе. Исходными данными для анализа являются изменения в прецедентах ИСП. Это означает изменение функциональных возможностей системы. Другое направление модификации - это изменение технологии (методик), например переход от централизованной архитектуры к распределенной или изменение графического интерфейса пользователя. Методики выполнения часто изменяются вместе с изменением бизнеса, но это не обязательно - может возникнуть потребность в их изменении без внесения изменений в бизнес-процессы.

Затем надо определить, какие из прецедентов старой системы будут затронуты изменениями. П-модель для изменяемых прецедентов строится на основе документации на существующую систему: спецификаций и требований, инструкций для пользователей; документации по сопровождению, обучающих материалов, спецификаций тестов и т.д.

Некоторые специалисты утверждают, что реинжиниринг можно автоматизировать полностью. Однако на практике оказывается, что для многих этапов работы этого сделать нельзя. В частности, вряд ли удастся обойтись без документации, представляющей собой наиболее удобное средство описания существующей информационной системы. Конечно, можно использовать простые средства генерации описаний кода. Но, к сожалению, это ничего не дает: в получаемых абстрактных описаниях трудно разобраться тем, кто не писал исходный код. Лучше всего здесь использовать руководство пользователя и спецификации тестов. Тестовые примеры помогут найти прецеденты информационной системы, а руководство пользователя - сформулировать описания полученных прецедентов.

Следующий шаг состоит в построении идеальной объектной модели для тех прецедентов, которые будут изменены. В ходе выполнения предыдущего и данного шагов понимание предполагаемых изменений информационной системы уточняется и становится возможным указать те части старой системы, которые надо сохранить и модифицировать.

Модели, построенные на предыдущих шагах, представляют абстрактный взгляд на существующую систему. Задача следующего шага - связать эти модели с существующей реализацией системы. Это означает, что необходимо отследить наличие взаимосвязей между понятиями в моделях и в проектируемых блоках ИСП. Исходными данными для этой работы являются, например, документация этапа реального проектирования, файлы с исходным кодом и описания схем баз данных для старой системы. Эту работу не удастся выполнить без помощи тех специалистов, которые хорошо знают программный код.

На данном шаге имеется достаточно сведений для определения интерфейса между старой системой и ее новой версией и создания детального описания интерфейсов между сохраняемыми и заменяемыми частями систем.

На этом этапе можно приступить к реализации перепроектированной системы, т.е. построить новую объектную модель, интерфейс и ввести необходимые изменения в исходную систему - и все это параллельно. Здесь потребуется провести анализ требований, идеальное и реальное проектирование (см. разд. 8.3). При реализации нового компонента информационной системы совсем необязательно использовать те же средства, что применялись при создании исходной системы. Можно выбрать другой язык программирования, например объектно-ориентированный. При подключении новых компонентов можно перейти к простейшей распределенной реализации, основанной на архитектуре "клиент-сервер".

Интерфейс, который создан для исходной и новой систем, должен обеспечивать совместное использование различных подходов. С его помощью новая объектно-ориентированная система должна иметь возможность рассматривать старую в терминах совокупности объектов, а старая - считать новую своим фрагментом, реализованным согласно прежним методикам.

Итак, реинжиниринг информационной системы есть следствие реинжиниринга бизнес-процессов. Прежде всего изменения требуют фрагменты, соответствующие интерфейсным и управляющим объектам в идеальной модели прежней системы. Объекты-сущности оказываются обычно более стабильными. Они соответствуют элементам данных (например, типы данных или записи) и программам (фрагменты процедур или подпрограмм), которые работают с этими элементами.

В первую очередь при проведении реинжиниринга исходной информационной системы следует сохранять, насколько это возможно, реализованную в ней модель данных и свести все необходимые изменения к минимуму. Если исходная система спроектирована не совсем плохо, не придется отказываться от всех данных, с которыми она работает. Обычно удастся почти полностью сохранить имеющуюся модель данных и, следовательно, большую часть самих данных. Основные изменения лучше всего направить на интерфейсные и управляющие объекты идеальной модели исходной системы. Скорее всего за счет развития и модификации именно этих объектов следует адаптировать информационную систему к новой организации бизнеса, поэтому с них и следует начинать.

8.5.2

Информационная технология (ИТ) как необходимое условие процесса инновации

ИТ сами по себе не обеспечивают то кардинальное обновление, которое ожидается получить от реинжиниринга. Интересно отметить, что результаты исследований влияния информационных технологий на бизнес вообще выглядят не очень впечатляюще. Детально развивая эту тему, Т.Давенпорт [3] отметил, что наиболее вероятная причина неудачных попыток проведения БПР состоит в неполном использовании возможностей информационной технологии.

Выполняя реинжиниринг бизнес-процессов, следует изучить, как различные ИТ могут обеспечить обновление вашего бизнеса, как они позволят достичь не просто ускорения, а радикального преобразования всей работы. Следует убедиться, что использование информационных технологий действительно повышает продуктивность процессов в бизнесе компании.

Какую же помощь могут оказать информационные технологии? В соответствии с идеями [3] можно выделить три категории изменений, которые обеспечивает использование информационных технологий.

Изменения, относящиеся к первой категории, позволяют улучшить временные характеристики процессов без модификации их содержания. Такие изменения, не являясь революционными, позволяют:

- автоматизировать работу и сократить ручной труд;
- анализировать данные новыми методами, которые невозможно применять вручную.

Вторая категория охватывает случаи реорганизации последовательности шагов по выполнению заданий в бизнес-процессе. Цель этой реорганизации состоит в существенном улучшении процессов обработки за счет:

- одновременного выполнения различных работ, используя базы данных и сети;
- перехода к распределенной организации данных, обеспечивающей доступ к информации из различных мест;
- вынесения части процессов за пределы компаний и предоставления клиентам или поставщикам возможности доступа к информационным системам;
- координирования действий, достигаемого за счет быстрого доступа к необходимой информации в пределах компании;
- использования экспертных систем для привлечения сотрудников средней квалификации к выполнению сложных высококвалифицированных работ.

Изменения из третьей категории не затрагивают сами процессы, но позволяют контролировать каждый конкретный экземпляр процесса и выявлять, где он наталкивается на те или иные проблемы. Еще один способ использования такой информационной поддержки состоит в измерении параметров функционирования процессов для выявления “узких мест”.

Несмотря на очевидность высказанных выше соображений, анализ показывает, что далеко не все компании реально отдают себе отчет в том, что они делают и как много времени они тратят на это. И уж совсем немногие компании достигли такой степени зрелости, при которой они действительно измеряют производительность процессов для того, чтобы решить, как и что следует реконструировать [5].

8.5.3

Инкрементное внедрение

Один из способов, обеспечивающих плавность установки новой информационной системы, состоит в инкрементном, то есть постепенном ее внедрении. Предположим, что построена П-модель идеальной информационной системы для обновленного варианта вашей компании. Эта модель основывается в большей мере на теории, а не на практике, поскольку она не готова и не может быть передана на тестирование будущим пользователям. Это означает, что нельзя быть уверенным в том, что модель правильно отражает потребности пользователей.

Один из способов, гарантирующих успешное решение проблемы, состоит в инкрементной установке новой информационной системы. Начинать следует с системы, реализующей лишь несколько прецедентов из П-модели информационной системы. Можно взять те прецеденты, которые поддерживают работу наибольшего числа сотрудников компании. Можно воспользоваться и другим критерием вы-

бора: реализовать те прецеденты, которые, как ожидается, дадут наиболее существенный выигрыш в сроках выполнения основных бизнес-процессов. Используя такие критерии, можно проранжировать прецеденты бизнес-системы с точки зрения их важности. Изменения, которые могут понадобиться для прецедентов с высоким рангом, имеют более высокий приоритет по сравнению с изменениями для прецедентов с низким рангом. Начиная пробную эксплуатацию информационной системы с поддержки наиболее важных прецедентов, вы снижаете риск ненужной работы, поскольку изменения, необходимые для второстепенных прецедентов, могут оказаться ошибочными или ненужными после внесения более приоритетных изменений.

Внедрив первый вариант новой информационной системы, можно перейти к работе над следующим. По мере накопления опыта реального использования системы будет формироваться новое, более точное представление о требованиях к ней. Появятся новые идеи относительно того, как лучше использовать информационную поддержку в компании, а это может заставить пересмотреть прецеденты бизнес-системы. В результате добавятся новые прецеденты использования информационной системы, какие-то прецеденты изменятся, а некоторые придется удалить.

Таким образом, внедрение информационной системы осуществляется инкрементно. Каждая итерация приводит к изменению представлений об идеальной информационной системе. Даже в наиболее благоприятном случае относительной стабильности этих представлений неизбежно происходит их уточнение, в результате которого достигается более глубокое понимание. Важно отдавать себе отчет в том, что информационные системы постоянно претерпевают изменения. Это прямой результат текущего изменения бизнес-процессов в ходе их реинжиниринга и совершенствования. Для построения информационных систем, приспособленных к постоянному развитию, лучше всего использовать объектно-ориентированные методы. Ни один из других доступных методов не позволит модифицировать системы так быстро, как этого требует современный бизнес.

Таким образом, цель этой главы состоит в том, чтобы показать естественность и простоту взаимосвязи и взаимодействия П-моделей и объектных моделей в описанном подходе. По П-модели бизнеса строится его объектная модель. Из объектной модели бизнеса получаются прецеденты информационной системы поддержки. При этом некоторые объекты-сущности бизнес-системы соответствуют объектам-сущностям информационной системы (рис. 8.19). Прецеденты информационной системы, в свою очередь, образуют базис для построения объектных моделей этой системы.

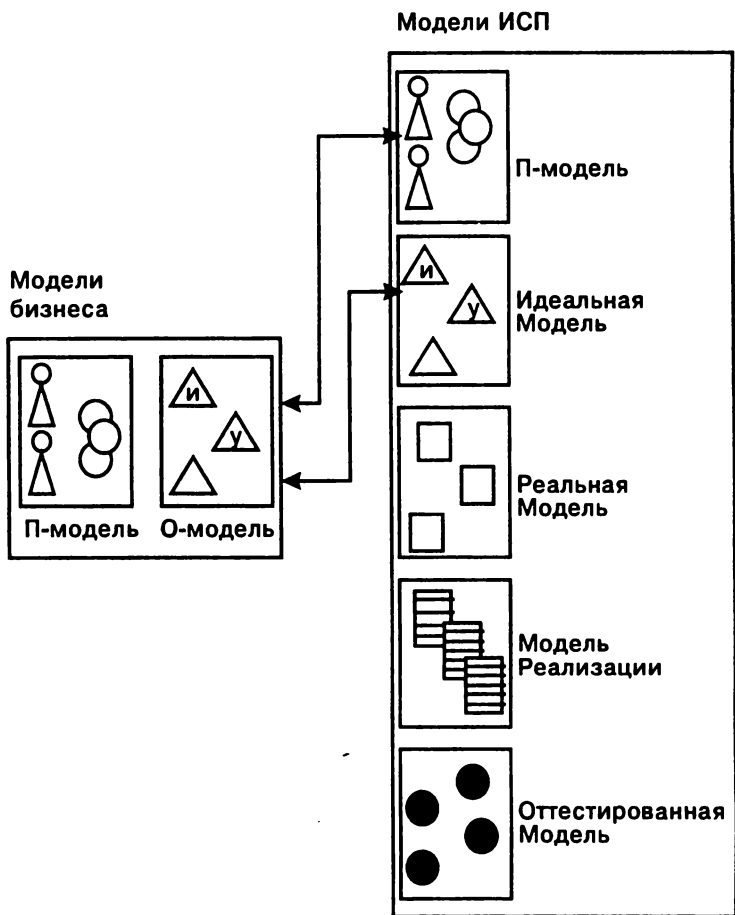


Рис. 8.19. Взаимосвязь между моделями бизнеса и информационной системы

Разработка программного и информационного обеспечения - это процесс создания последовательности моделей информационной системы. Каждая модель описывает некоторый аспект системы. Необходимость наличия нескольких моделей обусловлена существованием различных участников информационной системы, каждый из которых заинтересован в своем видении и соответствующей степени детализации описания системы.

ЛИТЕРАТУРА

1. Новоженев Ю.В. **Объектно-ориентированные технологии разработки сложных программных систем.** /Под ред. Е.Г.Ойхмана. - М: 1996.
2. Constantine L.L. **More than Just a Pretty Face: Designing for Usability**// Proceedings of Software Development, Spring Conference 1994. San Jose, CA, March, 1994.
3. Davenport T.H. **Process Innovation, Reengineering Work through Information Technology.** - Boston, MA: Harvard Business School Press, 1993.
4. Hammer M. and Champy J. **Reengineering the Corporation: A Manifesto for Business Revolution.** - N-Y: HarperCollins, 1993.
5. Humphrey W.S. **Managing the Software Process.** - Reading, MA: Addison-Wesley, 1989.
6. Jacobson I., Ericsson M., Jacobson A. **The Object Advantage. Business Process Reengineering with Object Technology**//ACM Press. - Addison-Wesley Publishing, 1995.
7. Jacobson I. **Basic Use-Case Modelling. Report on Object Analysis Design**// SIGS Publications, Inc. 1(2), July-August, 1994.
8. Jacobson I., Christerson M., Jonsson P. and Overgaard G. **Object-Oriented Software Engineering//A Use Case Driven Approach.** Reading, MA: Addison-Wesley; N-Y: ACM Press, 1992.
9. Jacobson I., Lindstrom F. **Reengineering of Old Systems to an Object-Oriented Architecture**//Proceedings of OOPSLA 1991. Phoenix, AZ, October, 1991.
10. Wirfs-Brock R., Wilkerson B. and Wiener L. **Designing Object-Oriented Software**//Englewood Cliffs. - N-Y: Prentice Hall, 1990.

9

МЕТОДОЛОГИИ И ИНСТРУМЕНТАЛЬНЫЕ СРЕДСТВА ДЛЯ ПРОВЕДЕНИЯ РЕИНЖИНИРИНГА

9.1

Методологии реинжиниринга

В настоящее время, кроме методологии, описанной выше, в гл. с 5 - 7, имеется несколько методологий БПР, разработанных различными консалтинговыми фирмами, и ряд инструментальных средств их поддержки. Однако после пятилетнего опыта проведения реинжиниринга специалисты сходятся во мнении, что не существует универсальной методологии, как не существует единого лекарства от всех болезней. В настоящее время ведущие консалтинговые фирмы располагают интегрированными совокупностями методов и приемов, а выбор методов определяется особенностями конкретного проекта по реинжинирингу. Чем шире диапазон методов, тем больше у проекта шансов на успех.

Реинжиниринг осуществляют специалисты двух типов - профессионалы в области реконструируемого бизнеса и разработчики информационных систем. Опыт реинжиниринга показывает, что по-настоящему успешное и новаторское внедрение информационных технологий является уникальным и творческим процессом: управляющие компаниями и специалисты-технологи, знакомясь с методами информационных технологий (ИТ), сами делают открытия относительно возможностей их использования в своем конкретном бизнесе [5, 7]. В то же время создание высококачественных информационных систем требует участия профессионалов в области ИТ. Возникает проблема нахождения общего языка. Решение этой проблемы стоит на пути интеграции таких современных технологий, как объектно-ориентированное программирование, CASE-технологии, инженерия знаний, имитационное моделирование процессов и средства быстрой разработки приложений [1].

Большинство современных консалтинговых фирм при проведении реинжиниринга используют CASE-средства. Здесь можно отметить такие известные фирмы, как Gemini Consulting (методология

Construct, инструментальное средство BusinessWorks, построенное в среде VisualWorks Smalltalk) и Andersen Consulting (методология Eagle и набор инструментариев). П.Хармон, рассматривая методологии этих фирм в своем обзоре [6], отмечает их ориентацию на профессионалов в области ИТ и направленность на разработку поддерживающих информационных систем.

Интересная методология предложена И.Якобсоном в его объектно-ориентированном подходе, основанном на прецедентах (use-case) [7]. Ранее И.Якобсон разработал подход, известный как объектно-ориентированный инжиниринг программного обеспечения. Отметив аналогию между конструированием технических систем, информационных систем и бизнес-процессов крупных компаний, И.Якобсон разработал собственную методологию параллельного создания двух взаимосвязанных систем - бизнес-системы и поддерживающей ее информационной системы компании.

В гл. 8 данной монографии описана суть подхода И. Якобсона. Создается последовательность моделей, описывающих обе системы как с точки зрения их использования (в первом случае - клиентами компании, во втором - пользователями информационной системы), так и с точки зрения их внедрения. При построении моделей используется общая методологическая база: модели первого типа описываются в терминах примеров использования (прецедентов), а модели второго типа раскрывают особенности реализации этих примеров в терминах объектно-ориентированного моделирования. Объектно-ориентированные модели описываются на различных уровнях детализации. Совместная разработка моделей обеих систем при общей методологической базе позволяет естественным образом учесть взаимосвязь этих систем и осуществить параллельное и согласованное их создание и последующее развитие. Для поддержки реинжиниринга разработана объектно-ориентированная программная среда разработки Objectory с элементами CASE-технологии. Методология И.Якобсона и среда Objectory взяты на вооружение рядом консалтинговых фирм и многими разработчиками инструментариев поддержки БПР. Однако модели, создаваемые в соответствии с этой методологией, довольно сложны, и маловероятно, что управляющие компаниями могут работать с ними так же естественно и легко, как профессионалы в области ИТ.

Еще один известный подход, предложенный Дж.Мartiном и Дж.Оделлом, был использован в ряде инструментариев, в том числе в системе OMW (Object Management Workbench) фирмы IntelliCorp. Его особенность состоит в сочетании CASE-технологии, объектно-ориентированного программирования и статических экспертных систем. Подход предусматривает создание диаграмм, представляющих потоки работ, структуры данных, взаимосвязи объектов, состояния и переходы в описании процессов. В отличие от всех предыдущих подходов здесь поддерживается процесс разработки программ-

ного обеспечения от диаграмм, описывающих модель бизнеса, до работающего кода. Этот подход, как и все предыдущие, ориентирован на разработчика информационных систем, а не на менеджеров компаний, в которых проводится реинжиниринг.

Как уже отмечалось (см. разд. 3.6), для обеспечения активного участия менеджеров в проведении реинжиниринга целесообразно интегрировать ключевые достижения современных информационных технологий - объектно-ориентированного программирования, CASE-технологий, имитационного моделирования процессов, инженерии знаний и средств быстрой разработки приложений. Именно такая тенденция и наблюдается в настоящее время в развитии методологий и инструментальных средств (ИС) БПР (см. также [2, 11, 12]).

Объектно-ориентированный подход в настоящее время признан базовой методологией БПР. Его особая роль объясняется следующим. Традиционно при создании информационных систем компаний разработчики отталкивались от данных. В результате используемые ими подходы к моделированию систем были ориентированы на описание данных о сущностях реального мира и их взаимосвязей, но не на поведение этих сущностей. Поскольку реинжиниринг ориентирован на процессы, а не на данные, традиционные подходы оказались неадекватны. Объектно-ориентированный подход (ООП) является в настоящее время единственным подходом, позволяющим описывать как данные о сущностях, так и их поведение. Кроме того, он обеспечивает создание "прозрачных", легко модифицируемых моделей бизнеса и информационных систем, допускающих повторное использование отдельных компонентов.

CASE-технологии использовались в реинжиниринге с самого начала его появления. Однако их ориентация на разработчиков информационных систем привела к тому, что в настоящее время их начинают объединять с другими современными технологиями, в первую очередь с объектно-ориентированными.

Имитационное моделирование обеспечивает наиболее глубокое представление моделей для непрограммирующего пользователя, а также наиболее полные средства анализа таких моделей. Модели создаются в виде потоковых диаграмм, в которых представлены основные рабочие процедуры компании и описано их поведение, а также информационные и материальные потоки между ними. Однако построение реальных имитационных моделей является довольно трудоемким процессом, а их детальный анализ (выходящий за рамки простого сбора статистики по срокам и стоимостям) зачастую требует от пользователя специальной подготовки. Для описания рабочих процедур может понадобиться дополнительное программирование. Таким образом, при попытке привлечь менеджеров к непосредственному использованию средств имитационного моделирования возникают определенные проблемы.

Чтобы преодолеть эти проблемы, в настоящее время начинают использовать методы *инженерии знаний*. С их помощью можно непосредственно представлять в моделях плохо формализуемые знания менеджеров о бизнес-процессах, в частности рабочих процедурах. Кроме того, решается проблема создания интеллектуального интерфейса конечного пользователя со сложными средствами анализа моделей.

Средства быстрой разработки приложений позволяют сокращать время создания поддерживающих информационных систем и, следовательно, необходимы не только в ходе реинжиниринга компании, но и на этапе эволюционного развития, сопровождающегося постоянными модификациями и улучшениями информационных систем компании.

В настоящее время переход к использованию интегрированных методологий и средств только начинается. В числе консалтинговых фирм, поддерживающих интегрированные методологии, можно указать компании Coopers & Lybrand (США) и Gensym (США). Методология, предложенная Coopers & Lybrand, основана на применении баз знаний о типовых бизнес-процессах, которые могут использоваться непосредственно менеджерами компаний. Компания на базе инструментального комплекса G2 фирмы Gensym разработала собственное инструментальное средство поддержки реинжиниринга SPARKS [8], что позволило ей объединить возможности объектно-ориентированного программирования, анимации и имитационного моделирования с CASE-технологией. Опыт фирмы Gensym в применении интегрированных ИС будет подробно рассмотрен в разд.9.3 и приложениях 1 и 2.

9.2

Классификация и анализ существующих ИС

Успех конкретного проекта по реинжинирингу во многом определяется используемым ИС. Важность правильного выбора ИС вызвана следующими обстоятельствами.

- Многие ИС автоматизируют только отдельные этапы БПР, что увеличивает риск неудачи и время проведения БПР.
- Большинство ИС ориентированы на программистов и не предполагают непосредственное участие менеджеров в разработке моделей компании, что зачастую приводит к неадекватности моделей и к непоправимым ошибкам при проведении БПР.
- Далеко не все ИС имеют выразительные средства, достаточные для того, чтобы адекватно описать деятельность конкретной компании. Например, практически все CASE-средства, используемые

при разработке моделей компаний, не имеют выразительных возможностей, позволяющих описать и промоделировать (если это необходимо) динамику процессов (см. разд. 5.1).

Современные инструментальные средства, используемые для проведения БПР, можно разделить на 5 категорий [9, 10].

❶ *Средства создания диаграмм и инструментарию низкого уровня* (Micrografx: ABC Flowcharter; Scitor: Process Charter; High Performance Systems: iThink). ИС этой категории являются дешевыми средствами (от 300 до 1000 дол.), предназначенными для автоматизации первой и, возможно, второй фазы реинжиниринга (рис. 9.1). Чаще всего используются для описания целей и перспектив компании. Не имеют связей со средствами быстрой разработки приложений; иногда включают элементы имитационного моделирования, но на довольно низком уровне.

Масштабы БПР

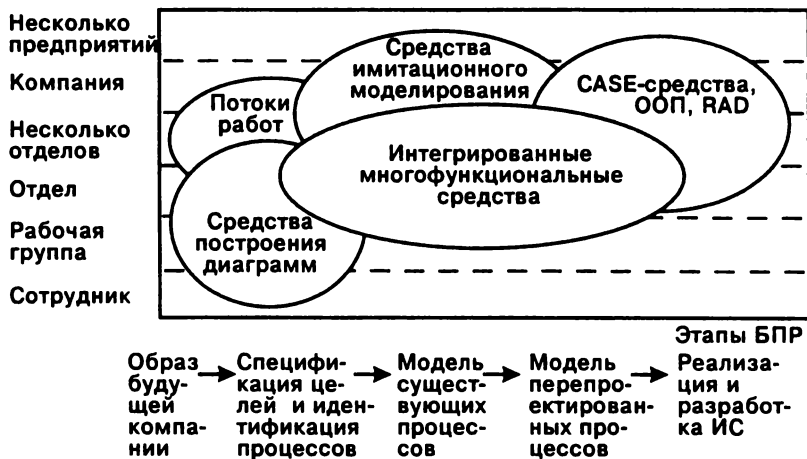


Рис.9.1. Категории ИС поддержки БПР

❷ *Средства описания потоков работ* (Action Technologies: ActionWorkflow Analyzer; Viewstar: Process Architect). Позволяют проектировать планы работы над проектами; просты в использовании, но средства анализа получаемых планов довольно слабые. Стоимость порядка 1000 дол.

❸ *Средства имитационного моделирования/анимации* (CASI: Modsim; Systems Modeling: Arena; ProModel: ProModel; Gensym: ReThink). Довольно дорогостоящие средства (от 10000 до 50000 дол.). Предлагают имитационное моделирование с помощью графич-

ческих средств, библиотек специализированных подпрограмм и специализированных языков; используются для выполнения сложных проектов, в крупных фирмах или на уровне нескольких организаций.

④ *CASE, объектно-ориентированные инструментари и средства быстрой разработки приложений* (Ptech: Framework, Oracle: Designer 2000; Popkin: Systems Architect). Многие разработчики CASE-средств и объектно-ориентированных средств начинают предлагать дополнения к своим инструментариям, позволяющие применять их в БПР. Эти инструментари ориентированы исключительно на разработчиков информационных систем.

⑤ *Интегрированные многофункциональные средства, автоматизирующие основные этапы проведения БПР* (Coopers & Lybrand: SPARKS; Meta Software: Workflow Analyzer; Protosoft Inc.: Paradigm; Interfacing Technologies: FirstStep; Texas Instruments Inc.: BDF; Gensym: ReThink + G2). Стоимость этих средств от 10000 до 50000 дол. Фирмы-поставщики предлагают методологическую поддержку, организацию многопользовательского доступа к инструментарию, стыковку со средствами быстрой разработки приложений и даже возможности имитационного моделирования и анимации. Использование этих средств требует специальной подготовки. Некоторые из этих средств (например, ReThink) ориентированы на бизнесменов и могут использоваться ими без посредничества специалистов в области ИТ.

В табл. 9.1 приведен перечень наиболее популярных инструментальных средств, используемых для проведения реинжиниринга.

Как показано на рис. 9.1, разные категории ИС, ориентированных на БПР, охватывают разные этапы реинжиниринга - от описания образа будущей компании до реализации процессов и генерации кодов информационной системы.

Дешевые средства создания диаграмм и описания потоков работ (категории 1 и 2) рекомендуется использовать в начале реинжиниринга, пока не выявлены особенности конкретного проекта и не получен материал, достаточный для принятия решения о закупке более дорогого многофункционального ИС. К достоинствам этих средств можно отнести их простоту и ясность, благодаря которым с ними могут работать непрограммирующие бизнесмены. Средства имитационного моделирования и анимации (категория 3) обеспечивают наиболее полный анализ динамики бизнес-процессов, а также "прозрачность" представления моделей бизнеса. При использовании этих средств сначала создаются детальные модели бизнес-процессов компании. Имитационные модели описывают не только потоки сущностей, информации и управления, но и различные метрики (например, частоту появления заявок, время выполнения каждой рабочей процедуры, возможно, с учетом случайных отклонений). Затем модели

**Инструментальные средства,
используемые для проведения реинжиниринга**

Продукт	Компания	Цена, дол.	Операци- онная система	ЭВМ	Коли- чество лицен- зий
ABC Toolkit	Micro-grafx Inc	895	Windows 3.1+	6MB VGA или 4MB RAM	> 40 млн
Activation Frame- work	The Real-Time In- telligent Systems	A.F.D.P.P. Windows -2495 NT-3495 UNIX-4995 A.F.-RT-GUI Windows-995 NT-1495 UNIX-1495	UNIX MS-DOS Windows NT Special orders	SUN SPARC SGI HP DEC DEC Alpha (OSF) SUN (OS& Solaris)	35
ARENA	Systems Modelling Corp.	PC Arena Pro- fessional-19000 PC Arena Simulation- 16000 PC Manufac- turing tem- plate-6000 PC Wafer Fab- rication Tem- plate-10000 WS Arena Pro- fessional-23000 WS Arena Simulation- 24000 WS Manufac- turing tem- plate-6000 WS Wafer Fabrication Template- 10000	DOS, OS/2 IBM HP DEC UNIX NOT WINDOWS!	PC UNIX Worksta- tion	>5000
Bpwin	Logic Works, Inc.	2495 Customer Sup- port Plan-425	Windows 3.x DOS 3.1	Intel 386, 486 или Pentium	10000

Продукт	Компания	Цена, дол.	Операци- онная система	ЭВМ	Коли- чество лицен- зий
Business Design Facility (BDF)	Texas Instruments	Single CPU license-10000	MS-WIN-DOWS IBM OS/2	Intel PC	
Business Improvement Facility	Virtual Software Factory		OS/2 UNIX ULTRIX AIX VMS		
DEC-model for Windows	DIGITAL Equipment Corp.		MS-DOS 5.0 MS Windows 3.0	Цветной монитор процессор 80386 4-8 MB	
Design/ IDEF & Workflow Analyzer	Meta Software Corp.	Design/ IDEF-3995 Workflow Analyzer-29995	MS-Windows UNIX/X Windows System Macintosh	HP9000 (700/800) для X Window System Sun Solaris для X Window System Macintosh Plus и выше	
Extend	Imagine That, Inc.	Extend single CPU-695 Extend+Manufacturing-990 Extend+BPR-990 Extend+BPR+Manufacturing-1285	Windows 3.1 Macintosh 6.07+	PC (386+), 8MB Macintosh, 4MB	
Factor/ AIM	Pritsker Corp.	US/cpu License-9500 GANTT Charts-1500	IBM OS/2 2.0 IBM Database 2 OS/2 (DB2/2)	IBM 80386 IBM 80486 min-10 MB Ram (+16MB) 10MB disk Math компю- рессор Рабочее диск. про- странство 40-60MB	5750

Продукт	Компания	Цена, дол.	Опера- ционная система	ЭВМ	Коли- чество лицен- зий
G2 (ReThink)	Gensym Corp.	15000-42000	Open VMS UNIX MS- Windows 3.1 Windows NT Windows 95	VAX Intel Based PC's Power PC Digital HP SUN IBM NEC Silcon Graphics Motorola	6000
GEM	Phoenix Systems Synectics Inc.	6995 (IBM OS/2, MS Windows) 15995-пакет из 5 500 (версия runtime) Annual Main- tenance-1500	Win- dows3.1+ Windows 95 Windows NT OS/2 2.0	IBM PS/2 + Compatibles Intel PC's	
GPSS/H	Wolverine Software Corp.	Personal-2500 Professional- 5200 simstat-1500 Unift II-975 Proof Anima- tion-3000 Proof Profes- sional-5200 Proof Anima- tor Demo Maker-2000	DOS 3.0+ Windows 3.x OS/2 2.x	PC VAX/VMS SUN SPARC Intel Compatibles 80386+Math процессор	3000
Ithink	High Per- formance Systems Inc.	715 969-(авторизо- ванная версия)	Macintosh System 7.0 Windows	Macintosh IBM Compatible	1000
LabView	National Instru- ments	Windows Full System-2655 Macintosh- 2655	DOS Windows SUN Macintosh	386/33 Math VCO 486/25 RAM: 8/4MB	
		SUN-5315	NEC 9800	Hard Disk: 4MB WIN- DOWS 3.1	

Продукт	Компания	Цена, дол.	Операци- онная система	ЭВМ	Коли- чество лицен- зий
Micro Saint	Micro Analysis & Design Inc.	DOS-2495 Windows-3995 Windows+Ac- tion View-5995 Macintosh- 3995 Macintosh+ Action View- 5995 UNIX-4750 UNIX+Human Performance Module-7195	MS-DOS или PC-DOS 3.0+ MS Windows 3.0+ Mac Plus или выше Silicon Graphics UNIX SUN	IBM PC или со- вместимый Macintosh Sun 4 + 4C Silicon Graphics	
Paradigm Plus	Protosoft Inc.	20000-30000	Microsoft Windows 3.1 и NT Sun SunOS 4.1.x Sun Solaris 2.x HP HP-UX 9.01 IBM AIX 3.2.x Silicon Graphics IRIX 5.x DEC Alpha OSF/1 AT&T NCR 3000	IBM Compatible Sun Station HP Work Station IBM Workstations (Power PC)	750
Pro Map	Invictus Systems Corp.	495	Windows 3.1 OS/2	PC+80486 процессор	
ProCap	Knowledge Based Systems, Inc.				
Process Charter For Windows	Scitor Corp.	595	DOS 5.0 или выше Windows 3.1	PC+80386 процессор и выше 4MB RAM (8MB) 8MB HD	

Продукт	Компания	Цена, дол.	Операци- онная система	ЭВМ	Коли- чество лицен- зий
Process Maker	Mark V Systems	1350 (single method) (PC) 950 (additional methods) 3350 (single method) (UNIX) 4000 (multimethod) (PC) 5000 mul- timethod) (UNIX)	MS Windows Windows NT UNIX	PC HP MIPS RS6000 SGI SPARC VMS ULTRIX и т.д.	
Process Modeller	ORACLE	2500	MS Windows	IBM Compatible	
Rapid Re	Antares Alliance Group	1195 119-Electronic Support	MS Win- dows 3.1 +	IBM Compatible 386+8MB RAM	
Service- Model	ProModel Corp.	13900	DOS 5.0+ MS Windows 3.1 Windows NT	IBM Compatible 486 процес- сор 12MB RAM 25MB дис- кового простран- ства 18MB для файла обмена	сотни
SES/work- bench	Scientific and Engi- neering Softw.	40000	SUN OS 4.1.3 Solaris 2.3 HP-UX 9.01- 9.05 или 8.07 ULTRIX 4.2 AIX 3.2.5	SUN SPARC station HP9000 (700), IBM RS/6000	
SIMFAC- TORY 11.5 - SIMPRO- CESS	CACI Product Company	PC OS/2-11500 PC Windows 3.1-15000 UNIX Work- station-15000	Windows 3.1 OS/2	PC OS/2 UNIX Workstations	тыся- чи

Продукт	Компания	Цена, дол.	Операци- онная система	ЭВМ	Коли- чество лицен- зий
SIM- SCRIPT 11.5	CACI Product Company		PC/DOS PC/OS2 Solaris VMS HPUX IBM AIX	IBM Compatible Sun Sparc Station HP RISC Macintosh IBM RS6000 VAX	10000
SIMU- LINK	Math Works Inc.	MATLAB (MS Windows или Mac)-1695 SIMULINK- 1995 Real-Time Workshop- 9995	MS Windows Macintosh OS UNIX VMS	DEC VAX 486 или Pentium Macintosh Sun SPARC DEC RISC DEC Alpha SGI HP9000 (700) IBM RS/6000	20000
SPARKS	Coopers & Lybrand	30000-60000	Open VMS UNIX MS Windows 3.1 Windows NT Windows 95	VAX Intel Based PC's Power PC Digital HP SUN IBM NEC Silcon Graphics Motorola	сотни
StP/BPR	IDE Canada	30000	UNIX SUN HP	SUN SPARC HP 9000 (700/800) IBM RS/6000 Digital Alpha OSF/1	15000 в 25 стра- нах

Продукт	Компания	Цена, дол.	Операци- онная система	ЭВМ	Коли- чество лицен- зий
System Architect/BPR	Popkin Software Systems	MS Windows Merge Version-1395	MS Windows OS/2 PM	IBM Compatible	50000
Professional	Inc.	Network Version-1545 OS/2 PM Merge Version-1795 Network Version-1995 SA/BPR Professional-1995			
Taylor II Simulation	F&H Simulation	15500	MS-DOS	80386/80486 2MB RAM VGA	
TeamFlow	CFM Inc.	295	MS Windows 3.1 Macintosh System 7 или позднее	PC 386+ Macintosh SE	
The Workflow Factory	Delphi Consulting Group	595	Windows 3.x	386 SX VGA 10MB/4MB RAM	
Thinking Tools	Thinking Tools Inc.	70000-100000-прототип проекта 600000-(дополнит.) полная имитация	DOS	зависит от проекта	
WITNESS	AT&T ISTEEL	5000-21000	Windows 3.0 или 3.1 OS/2 1.1, 1.2, 1.3, 2.0 VMS 5.4 или позднее SUNOS 4.1.1 SOLARIS 2.0 или 2.1 HP-UX 8.05/8.07 AIX 3.2	IBM-PC или совместимый 386/486 Processor VAXStation 3100 VAX 4000 VLC, 6000, 8000, 9000 SPARC2 (IPC/IPX/690) HP9000 (700/800) RS6000	

Продукт	Компания	Цена, дол.	Операци- онная система	ЭВМ	Коли- чество лицен- зий
Wizdom Works!	WIZDOM Systems Inc.		MS Windows	IBM Compatible	
Workflow Analyzer	Meta Software Corp.	29995 5000- за ежегодное сопро- вождение	UNIX/X Windows Systems Macintosh	HP9000 (700/800) для X Windows System Sun Solaris для X Windows System Macintosh Plus или выше	
Workflow BPR	Holosofx Inc.	2495	Windows	IBM Compatible	

“проигрываются” в сжатом времени или пошаговом режиме. При отсутствии анимации модели могут создаваться графически или аналитически. Если есть анимация, то модели представляются в виде диаграмм процессов; в ходе “проигрывания” модели эти диаграммы, очереди, а также поведение системы в целом визуализируются. Благодаря этому пользователь может получать полное представление о работе исследуемой системы.

Развитые средства имитационного моделирования пришли в БПР из промышленности и космических исследований. Работа с ними требует от пользователей определенной математической подготовки, поэтому на практике поставщики подобных средств всегда предоставляют консалтинговые услуги. “Прозрачность” представления моделей, возможность глубокого их изучения (особенно в случае сложных ответственных проектов) делают методы имитационного моделирования и анимации одним из перспективных направлений БПР. Возможности имитационного моделирования (в том или ином объеме) включают и в инструментарий, относящийся почти ко всем категориям ИС БПР. Увеличивается число фирм, использующих методы имитационного моделирования при проведении реинжиниринга.

Включение методов БПР в традиционные средства разработки программного обеспечения (ИС категории 4) представляется нетривиальной задачей. Если базой для моделей, создаваемых в ходе ре-

инжиниринга, являются процессы, то традиционный инжиниринг программного обеспечения основывается на данных. Наилучшим решением проблемы оказалось использование объектно-ориентированного подхода к разработке программного обеспечения, который позволяет описывать в объектах как данные, так и поведение (процессы).

Интегрированные многофункциональные средства (категория 5) поддерживают наибольший объем функций, используемых при проведении БПР. Многие из них обеспечивают хорошую методологическую поддержку, средства имитационного моделирования, модульность, средства коллективного доступа к моделям и нередко стыковку со средствами быстрой разработки приложений - RAD. Поэтому при реализации больших проектов по реинжинирингу рекомендуется использовать именно эти средства. Все ИС, относящиеся к категории 5, имеют хорошие средства спецификации процессов (поддержка методологии IDEF, потоки работ в сочетании с объектной ориентацией и т.д.) и средства оценивания процессов. Некоторые из рассматриваемых ИС включают средства имитационного моделирования. Реализованы, хотя и не в полной мере, средства оценивания по рабочим процедурам. Почти во всех ИС имеются средства анализа критического пути, присутствует методологическая поддержка. Многие ИС поддерживают режим коллективной разработки моделей. Предусматривается стыковка со средствами разработки приложений. По степени реализации перечисленных функций наиболее удачными являются такие системы, как ReThink и G2 (Gensym), Workflow Analyzer (Meta Software), SPARKS (Coopers & Lybrand) и Process Wise (ICL). Однако лидер в этой группе средств еще не определился.

В настоящее время идет активное развитие интегрированных многофункциональных ИС в части простоты использования, полноты средств имитационного моделирования, стыковки со средствами разработки приложений и т.д. В этом секторе особый интерес представляет разработка фирмы Gensym - инструментального средства ReThink [1] (см. разд. 9.3) и средства SPARKS [8]. В этих системах объединены возможности ключевых современных информационных технологий: графический объектно-ориентированный язык для описания моделей и проектов, средства анимации и имитационного моделирования реконструируемых процессов, методы искусственного интеллекта для полного и адекватного представления экспертных знаний о процессах. Все это открывает доступ к непосредственному моделированию и реконструированию бизнес-процессов новой группе пользователей - менеджерам. Сочетание "прозрачных" средств интерактивной графики с мощными возможностями моделирования процессов в реальном времени позволяет им самостоятельно, без помощи программистов воплощать свои идеи в виде работающих моделей процессов.

Оба средства реализованы на G2 (см. Приложение 1) и подобны по выполняемым функциям. В следующем разделе приведено описание ReThink. Мы отдали ему предпочтение перед SPARKS, так как он был реализован на 6 лет позже (в 1994 г.) и в нем использованы более современные блоки, функции и интерфейсы.

9.3

ReThink - инструментальный многофункциональный комплекс для проведения реинжиниринга

9.3.1

Введение в систему ReThink

ReThink – это объектно-ориентированное средство графического моделирования и проектирования бизнес-процессов. Он предлагает графическую среду для тестирования моделей, а также для определения различных параметров процесса, таких, как время, стоимость и др. ReThink позволяет динамически изменять изображение моделируемых процессов, что обеспечивает наглядность и понятность процессов любой сложности.

Система ReThink построена на базе инструментального комплекса G2 (см. Приложение 1). Таким образом, она является проблемно-ориентированным приложением комплекса G2, которое позволяет разработчикам использовать не только специализированные средства моделирования бизнес-процессов, но и универсальные средства комплекса по созданию интеллектуальных объектно-ориентированных систем управления реального времени.

Для представления моделей бизнес-процессов используются диаграммы, состоящие из блоков и соединений. Блоки представляют задачи в бизнес-процессах, а соединения - потоки сущностей: документов, информации, а также предметов, фигурирующих в бизнесе (например, запасных частей или упаковок с отпускаемой продукцией). В системе реализован ряд стандартных блоков, которые могут быть использованы в качестве сборочных элементов для построения работающих моделей любых процессов, например: источник заявок, принятие решения, обработка задания. Свойства и поведение блоков могут описываться как точными, так и случайными величинами. В случае необходимости разработчик может переопределять поведение блоков или задавать новые их классы с помощью базовых средств комплекса G2.

Объектная ориентация системы ReThink позволяет создавать понятные и наглядные модели бизнес-процессов, что существенно упрощает освоение и использование системы непрограммирующими пользователями. Объекты, построенные в результате моделирования бизнес-процессов, являются естественной основой для проектирования информационных систем поддержки этих процессов. В этом смысле средства системы ReThink могут рассматриваться как развитие CASE-средств. ReThink поддерживает анимацию потоков работ в ходе моделирования деятельности компании. Благодаря этому менеджер имеет возможность непосредственно наблюдать функционирование моделей, что повышает степень его доверия к результатам моделирования.

ReThink включает ряд базовых компонент, на основе которых строится модель бизнес-процессов:

- Сценарии управляют механизмами моделирования дискретных событий и позволяют проводить одновременное исполнение нескольких моделей.

- Блоки выполняют операции над объектами, такие, как создание объектов, исполнение бизнес-функций, установление и разрыв ассоциаций между объектами, удаление объектов.

- Инструменты позволяют получать и отображать в числовой и графической форме данные о производительности моделируемых процессов, а также позволяют вводить параметры в заданных точках модели.

- Ресурсы (средства труда) предназначены для ограничения исполняемых операций на основе объема и состава наличных ресурсов.

- Рабочие объекты (предметы труда) проходят через блоки модели и обрабатываются ими, аккумулируя статистики производительности в каждой точке моделируемого процесса.

ReThink поддерживает создание иерархических моделей, позволяющих описывать процессы с различной степенью детализации. Это обеспечивает простоту и естественность при создании сложных моделей больших компаний (рис. 9.2). Все элементы моделей, включая ресурсы процессов, могут модифицироваться непосредственно во время исполнения. Результаты изменений можно увидеть сразу же после их введения.

ReThink позволяет формировать стоимостные и временные характеристики различных проектов для объективного их сравнения, а также проверять гипотезы “Что, если”. Для анализа работы моделей предусмотрен набор инструментов: “зонды” для сбора данных и “установщики”, устанавливающие значения атрибутов сущностей; графики для наглядного отображения результатов моделирования; всевозможные просмотревые табло из стандартных средств комплекса G2. С помощью зондов можно снимать такие показатели, как длительность цикла обработки сущности на том или ином этапе, стои-

Модель верхнего уровня



Пулы ресурсов



Подмодель "Доставка"

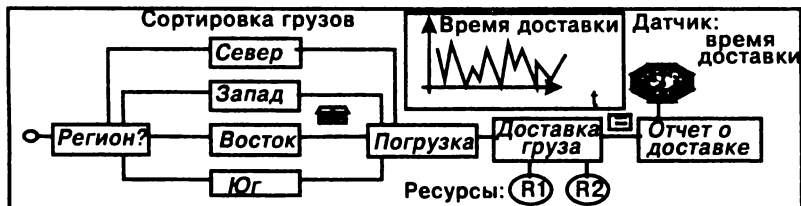


Рис. 9.2. Пример иерархической модели в системе ReThink

мость обработки, а также любые другие свойства, определенные разработчиком модели. Для отсева шумов и выявления тенденций можно использовать специальные блоки-фильтры.

Для проверки гипотез "Что, если" в системе реализован механизм сценариев. Сценарии позволяют исследовать зависимость поведения одной и той же модели от поведения внешнего мира (например, частоты поступления заявок, сложности этих заявок и т.д.) и каких-либо параметров этой модели (например, количества транспортных средств или численности служащих, занятых оформлением заказов). Варьируемые параметры и измеряемые показатели выносятся на отдельное окно сценария, после чего в результате прогона модели автоматически формируется отчет. Кроме этого ReThink позволяет использовать сценарии для объективного сравнения альтернативных проектов: один и тот же сценарий, описывающий некоторое заранее заданное поведение внешнего мира, может использоваться для прогона различных моделей. Результаты прогона, вынесенные в отчет, являются основой для сопоставления и оценки этих моделей.

ReThink поддерживает коллективную работу с приложениями на основе архитектуры "клиент-сервер" с помощью клиентной системы Telewindows комплекса G2. Коллективная разработка и использование приложений имеют принципиальное значение при проведении глобального реинжиниринга крупной компании или объединения, например нескольких компаний в рамках отрасли.

Как и инструментальный комплекс G2, система ReThink функционирует на большинстве рабочих станций в среде UNIX, системах OPEN VMS, а также на Pentium PC в среде Windows NT, Windows 95

и Windows 3.1. Система Telewindows позволяет обращаться к системе с персональных компьютеров Intel 386/486 в среде Windows 3.1. Экранные сообщения ReThink русифицированы, т.е. пользователь общается с ReThink на русском языке.

При создании системы ReThink фирма Gensym не ставила своей целью предложить какую-либо конкретную методологию реинжиниринга. Ее задача - создание удобного универсального средства для реализации различных методологий. Система ReThink адресована в первую очередь консалтинговым фирмам и информационным подразделениям крупных компаний для воплощения их оригинальных идей в области реинжиниринга. Система предоставляет возможность развития средств инструментального комплекса G2 - вплоть до реализации новых нестандартных средств моделирования и анализа. Особый интерес представляет создание проблемно-ориентированных баз знаний о типовых бизнес-процессах.

Система ReThink успешно используется в ряде компаний, в том числе в патентном ведомстве США и компании Xerox, проводившей реинжиниринг отделения по закупке сопутствующих материалов с годовым оборотом в 3 млрд дол. В компании Xerox при проведении реинжиниринга сначала использовался пакет ABC FlowCharter. Построенная модель работы отделения включала 17 процессов и 314 рабочих процедур. Анализ модели показал, что 70 % процедур оказались непроизводительными. Затем была разработана новая модель процессов закупки, включающая всего 42 рабочие процедуры. Столкнувшись с таким существенным сокращением количества процедур, руководство компании поставило вопрос о работоспособности новой организации: не встанут ли перед компанией серьезные непредвиденные проблемы после того, как она сделает основные капиталовложения в реконструкцию отделения? Чтобы обосновать предложенный проект, было решено использовать систему ReThink, с помощью которой предполагалось исследовать имитационную модель предлагаемой организации работы отделения. В результате несколько процессов пришлось снова перепроектировать. Таким образом, использование мощных средств моделирования привело к явному выигрышу в качестве проекта, следовательно, снизило риск неудачи при проведении реинжиниринга.

9.3.2

Детальное описание системы ReThink

ReThink выполняется в среде разработки G2. Следовательно, в начале каждого сеанса разработки модели необходимо загрузить G2. В начале работы ReThink открывает свое меню верхнего уровня. По

умолчанию в нем доступен единственный пункт ReThink: Gensym's BPR Toolkit. Выбрав его, вы получите меню с тремя пунктами:

Пункт меню	Описание
File	Содержит команды, позволяющие показать стартовое окно и вернуться к меню верхнего уровня ReThink. Позже здесь появятся команды загрузки и сохранения ваших моделей.
Palettes	Содержит команды доступа к палитрам встроенных блоков, инструментариев, инструментов и ресурсов, используемым при построении модели.
Application	Первоначально не содержит никаких команд. К этому меню можно добавлять свои команды, например команды доступа к отдельным фрагментам разрабатываемой модели.

На рис. 9.3 изображены палитры блоков (blocks), инструментариев (tools), инструментов (instrumentation) и ресурсов (resources).

Использование сценариев. Инструментарии ReThink являются средствами, упрощающими процесс организации и управления создаваемой моделью. Основными инструментариями в ReThink являются: Модель (Model tool) и Сценарий (Scenario tool).

Сердцем ReThink является его средство моделирования на базе дискретных событий, которое отслеживает события по таймеру моделирования. Этим таймером можно управлять при помощи *сценариев* ReThink. Инструментарий Сценарий позволяет управлять различными аспектами модели и отображать следующие параметры:

- состояние модели, т.е. запущена ли она, остановлена или временно приостановлена;
- режим, в котором работает модель, например режим моделирования или режим отладки;
- ключевые входные параметры модели, например стоимость ресурсов;
- итоговую статистику и показания, вычисляемые моделью;
- графическое представление модели в виде графиков и диаграмм.

Таким образом, сценарии служат центром управления данной моделью, позволяя облегчить проведение анализа “Что, если” в реорганизации бизнес-процессов.

Инструментарий Модель используется для организации создаваемой модели. В процессе разработки модели выполняются различные эксперименты с альтернативными сценариями, производится тестирование модели и проводятся эксперименты с альтернативными процессами.

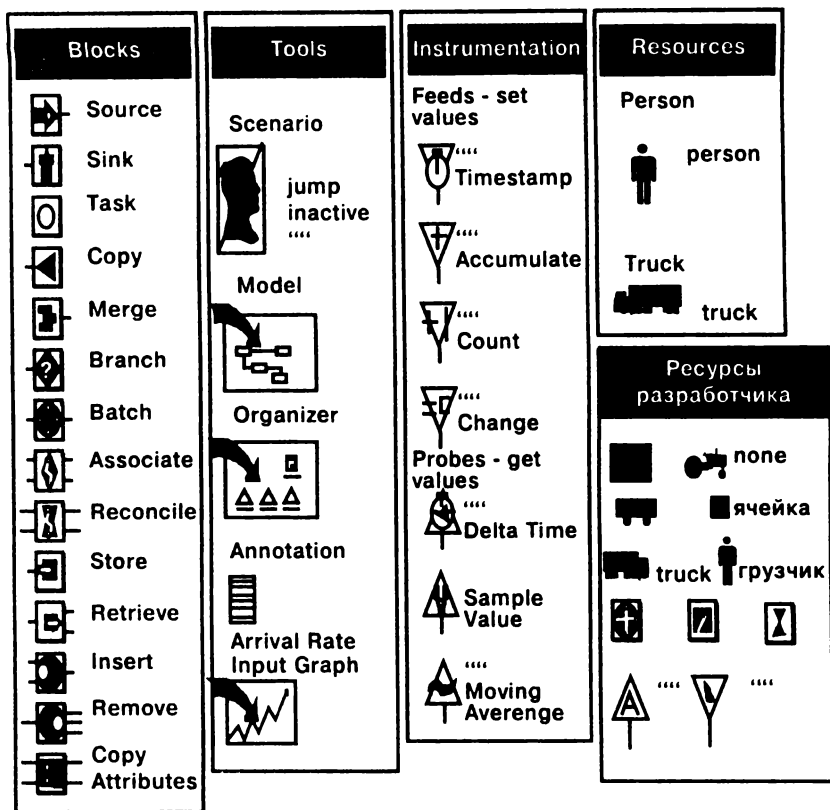


Рис. 9.3. Базовые компоненты системы ReThink

Перед построением модели необходимо определить, как будет организована информация. Один способ – разрабатывать модель непосредственно на рабочем пространстве и создавать иерархию рабочих пространств. Другой способ – создавать модель с помощью инструментария Модель.

Инструментарий Model имеет подпространство, на котором создается модель. Для каждой модели в ReThink требуется связанный с ней инструментарий Сценарий. Как правило, сценарий находится на том же рабочем пространстве, что и инструментарий Модель, или на рабочем пространстве выше по иерархии.

Сценарий имеет три атрибута: режим (mode), статус (state) и время моделирования (clock time).

Имя Mode (режим)	По умолчанию jump (дискретный)	Описание
		Указывает текущий режим выполнения сценария. Имеются следующие режимы выполнения сценария: jump, step и synch (дискретный, пошаговый и синхронный). Они описываются ниже.
State (статус)	inactive (неактивен)	Указывает текущий статус сценария. По умолчанию – неактивен, что означает, что сценарий не запущен. Активный сценарий может иметь четыре других состояния: stopped, reset, running и paused (остановлен, переустановлен, запущен и приостановлен). Они описываются ниже.
Clock time (время)	“ “	Указывает текущее время моделирования. Значение по умолчанию – пустая строка. При сбросе сценария ReThink показывает фактическое текущее время – время начала моделирования. Стартовое время сценария можно редактировать.

Чтобы запустить модель, необходимо активизировать связанный с ней сценарий. Это можно сделать либо с помощью команд меню, либо с помощью кнопок на подпространстве сценария.

По умолчанию сценарий неактивен, что и показывает подпись к пиктограмме сценария. Открыв меню сценария, мы получим те пункты меню, которые видимы на основании текущего состояния. Выбор пункта activate scenario приводит к тому, что сценарий становится активным, но не запущенным на выполнение. При этом пиктограмма сценария становится синей, а статус сценария изменяет свое значение на stopped.

ReThink показывает панель управления сценарием (Scenario Control Panel). Это рабочее пространство содержит несколько кнопок для управления сценарием. Подпространства сценария – это окна, где размещаются поля ввода, позволяющие конечному пользователю задавать входную информацию для модели, а также просмотрные табло и графики, выдающие сводную информацию о ходе моделирования.

Активизировав сценарий, его можно запускать, приостанавливать, переустанавливать или останавливать процесс моделирования. Как правило, статус сценария изменяется с помощью кнопок на подпространстве сценария или же с помощью команд меню сценария. Текущий статус отображается на пиктограмме сценария.

Изменение режима выполнения сценария. Изменяя режим выполнения сценария, изменяют режим моделирования.

В ReThink предусмотрены три режима выполнения сценария:

Режим	Описание
Jump (дискретный)	Обычный дискретный режим моделирования. При работе модели происходят различные события. ReThink обрабатывает эти события в порядке их поступления. В этом режиме ReThink после обработки каждого события автоматически переводит таймер на время начала очередного события.
Step (пошаговый)	Режим, используемый для диагностики и отладки. В пошаговом режиме ReThink приостанавливается после каждого события, чтобы дать возможность выполнить процесс моделирования по шагам. После продолжения моделирования таймер автоматически переводится на время начала следующего события. В этом смысле режим Step подобен режиму Jump.
Synch (синхронный)	Режим, используемый для визуализации задержек в процессе. В синхронном режиме ReThink отображает реальное время на модельное время в некотором масштабе. Например, в этом режиме можно запустить моделирование со скоростью один час реального времени в секунду. Это позволяет сосредоточиться на анализе информационных потоков, предоставив таймеру отслеживать время моделирования автоматически.

Режим выполнения изменяется с помощью кнопок на подпространстве сценария или посредством команд меню. Текущий режим указывается на пиктограмме сценария.

Переключение между режимами клиентов. Существуют три типа клиентов ReThink. Клиенты каждого типа работают в своем режиме:

- конечные пользователи модели работают в режиме Пользователь (User);
- разработчики модели работают в режиме Разработчик (Developer);
- системные администраторы работают в режиме Администратор (Administrator).

В следующей таблице приводится описание этих трех режимов.

Режим	Описание
User	Клиенты приложений ReThink – менеджеры и рядовые сотрудники – работают в режиме User. Они используют уже разработанные модели для осуществления анализа “что, если” по различным сценариям. В режиме User можно изучать модель, не рискуя ее каким-либо образом повредить.
Developer	Этот режим, как правило, используется для проектирования и построения моделей процессов. Некоторые ограничения этого режима позволяют скрыть ненужные внутренние детали работы ReThink.

Administrator Этот режим используется, если требуется изменить внутреннее поведение ReThink. Предполагается, что функции ReThink могут настраиваться лицами, осуществляющими консалтинговые услуги в реорганизации бизнес-процессов.

Создание модели. Первым шагом создания модели является копирование блоков из палитры и соединение их между собой. Все блоки ReThink размещаются на палитре Блоки (см. рис. 9.3). Графические объекты на этой палитре являются фундаментальными строительными блоками для моделирования деталей любого процесса. Они независимы от проблемной области; их можно использовать для любой отрасли. Можно изменить их внешний вид и поведение, чтобы они адекватно отражали конкретный рынок или компанию.

При проектировании процессов лучше начать с самого верхнего уровня, выделить входы и выходы для каждого процесса и затем постепенно переходить к деталям. ReThink способствует применению этого подхода, позволяя создавать подпространства для блоков, увеличивая степень детализации.

Блоки. В ходе моделирования бизнес-процессов выполняются различные задачи, которые оперируют с различными видами объектов. Для представления каждого шага процесса используются блоки ReThink (см. рис. 9.3).

Возможность простого манипулирования графическим представлением блоков в G2 (см. Приложение 1) и составления схем, являющихся отображением технологических цепочек или абстрактных алгоритмов обработки данных, обеспечивает базовые средства для построения проблемно-ориентированных языков визуального программирования. В данном случае блоки приобретают свойства операторов и в совокупности с различными классами связей формируют грамматику нового языка. Основным преимуществом такого подхода является то, что сформированная диаграмма потоков информации и есть исполняемая программа, промежуточные фазы генерации кода и компиляции для ее использования не требуются. Впервые эта концепция была реализована фирмой Gensym в GDFL - языке графического представления информационных потоков для построения систем диагностики реального времени в системе GDA (G2 Diagnostic Assistant - см. Приложение 1).

Подход оказался настолько удачным, что используется теперь во всех проблемно-ориентированных расширениях G2. Для представления типовых задач ReThink предлагает репозиторий с настраиваемым поведением. Для удобства использования блоки объединены в общей палитре. Если разработчика не устраивает набор предлагаемых блоков, можно создавать новые, группируя их из существующих или описывая их поведение. Пиктограммы блоков, соединенные графическими связями, отображают последовательность выполнения и взаимодействие между существующими с точки зрения управления задачами.

Блоки оперируют с объектами: создают их, уничтожают, копируют, устанавливают между ними связи, собирают их, разбирают и т.д. Блоки используются для задания поведения бизнес-процесса. Блоки создаются путем копирования их из палитры и помещения на рабочее пространство. Чтобы предоставить пользователю текстовую информацию о назначении блоков, рекомендуется присваивать им имена (метки).

Входами модели являются блоки-источники, генерирующие с определенной интенсивностью рабочие объекты. Обработка каждого из блоков других типов инициируется поступлением рабочего объекта от предыдущего блока. Объекты передаются по путям (*bpr-path*), формирующим последовательность выполнения задач. Процесс обработки блока состоит из трех фаз: планирования; начала активности; завершения активности.

В течение фазы планирования проверяются готовность каждого блока к выполнению своей задачи, наличие всех необходимых входных рабочих объектов и доступность используемых ресурсов. Если все эти условия выполнены, создается объект, называемый “активность” (*activity*), и блок переходит в фазу активности. На этой фазе над рабочими объектами выполняются “входные” операции, определяемые данным блоком. Эти операции задаются в методе *bpr-start-method* для каждого класса блоков. После этого рассчитываются стоимостные и временные показатели блока и рабочих объектов. Во время фазы завершения активности над объектом выполняются “выходные” операции, освобождаются задействованные ресурсы, вычисляются значения величин, измеряемых инструментами данного блока, и рабочие объекты передаются следующим блокам в технологической цепочке. Завершающие операции задаются методами *bpr-stop-method*. Для индикации выполнения каждой из стадий процесса изменяются цвета отдельных областей пиктограммы блоков, ресурсов и рабочих объектов.

Пути *ReThink* задают последовательность выполнения операций обработки блоков и содержат очереди рабочих объектов, ожидающих обслуживания блоками. Кроме того, они содержат атрибуты, используемые в операциях принятия решения (см. ниже Блок ветвления), дают возможность изменять дисциплину обслуживания рабочих объектов блоком, а также используются для анимации.

Опишем назначение каждого блока из палитры Блоки (*Blocks*) (см. рис. 9.3).

Блок *Source* (источник) служит для создания объектов, являющихся входными для процесса. Например, можно использовать блок *Source* для создания заявок, звонков и любых других «внешних раздражителей», поступающих в компанию. Временные характеристики для блока *Source* можно задавать с помощью атрибутов, файла или графика.

Блок Sink (приемник) работает в паре с блоком Source. Блоки типа Sink размещаются в конце процесса. В ReThink их использование не требуется, но они полезны для обозначения конечной точки какой-либо линии обработки.

Блок Task (задача) обозначает любой вид деятельности, приносящий добавленную стоимость. Он может иметь любое число входов и выходов. Блок может представлять элементарную операцию и набор операций, т.е. подпроцесс. Блок Task можно последовательно декомпозировать.

Блок Copy (копирование) служит для создания нескольких экземпляров объекта. Блок Copy может иметь любое число входов и выходов. При поступлении объекта на вход блок Copy создает на выходе столько же экземпляров этого объекта, сколько у него выходных путей.

Блок Merge (слияние) служит для слияния двух или более потоков. Он может иметь любое число входов, но только один выход. Получая объект на входе, блок Merge передает его на выход. Этот блок можно использовать, например, для объединения двух отдельных потоков заявок.

Блок Branch (ветвление) служит для реализации любой операции, требующей принятия решения, например выбора маршрута, сортировки, сравнения или любого другого вида ветвления. Блок Branch поддерживает несколько видов принятия решений. Например, можно промоделировать принятие решений на вероятностной основе. Ветвление также можно осуществлять на основании типа поступающего на вход объекта, на основании значений атрибутов объекта или на основании любого другого пользовательского критерия. Блок Branch универсален; он может промоделировать любое ветвление.

Блок Batch (упаковка) служит для группирования объектов в один пакет. ReThink ожидает, пока на входе блока не наберется достаточное количество объектов, и затем передает их на выход.

Блок Associate (ассоциация) служит для создания связи между двумя объектами, которые обрабатываются раздельно. Например, с помощью блока Associate можно установить связь между заявкой и соответствующим ей счетом, чтобы впоследствии объединить их в ходе обработки. Объединение связанных объектов производится с помощью блока Reconcile.

Блок Reconcile (ожидание пары) служит для объединения объектов, связь между которыми установлена блоком Associate. Например, с помощью этого блока можно объединить заявки и счета в пары. При этом заявки и счета будут формировать очереди на входе, ожидая своего напарника. Когда на вход поступает счет, связанный с одной из заявок в очереди, блок Reconcile одновременно передает на выход и заявку, и счет, направляя объект каждого типа по своему каналу. Далее в ходе обработки эти объекты можно удалить, вместе занести в базу данных, вставить один в другой и т.п.

Блоки Store и Retrieve (сохранение и извлечение). Блок Store служит для моделирования любых операций базы данных. Он помещает входные объекты в буфер, где они ожидают извлечения с помощью блока Retrieve. Например, блок Store можно использовать для моделирования базы данных по клиентам, базы данных по заказам или склада производственного оборудования. Это универсальный блок, с его помощью можно смоделировать любой участок процесса, представляющий собой независимое хранилище.

Блоки Insert и Remove (вставка и удаление). Блок Insert служит для вставки одного объекта в другой, а блок Remove – для удаления вставленного объекта. Например, с помощью этих блоков можно добавить к заказу несколько объектов-строчек и удалить их в дальнейшем.

Блок Copy Attributes (копирование атрибутов) служит для копирования значений атрибутов, определенных пользователем, из одного объекта в другой.

Конфигурирование модели. При конфигурировании моделей, как правило, выполняются два основных шага:

- указываются рабочие объекты, которые будут создаваться и обрабатываться моделью;
- указываются временные характеристики событий процесса.

Модель в ReThink создает и обрабатывает объекты, называемые **рабочими объектами**. Тип рабочего объекта, который будет обрабатываться блоком, указывается в атрибуте *type* выходного пути данного блока.

Каждый блок в ReThink имеет ряд атрибутов, некоторые из которых являются стандартными для G2:

Атрибут	Описание
notes	Атрибут G2, описывающий статус блока.
names	Атрибут G2, задающий уникальное имя для блока.
label	Метка ReThink, позволяющая идентифицировать блок. Этот атрибут не идентичен атрибуту <i>names</i> .
error	Атрибут G2, описывающий ошибки, относящиеся к данному блоку.
comments	Атрибут G2, содержащий текст комментария.

Кроме того, блок имеет три атрибута, являющихся подобъектами. Эти подобъекты содержат атрибуты, позволяющие настраивать описываемые ниже параметры блока.

Подобъект	Описание
animation-subtable	Задает процедуру, определяющую анимацию данного блока и его цвет в следующих состояниях: активен, неактивен, находится в состоянии ошибки.

Подобъект	Описание
duration-subtable	Задаёт процедуру, определяющую временные характеристики обработки в блоке: общее время обработки объекта, общее истекшее время, среднее время обработки, различные временные расчетные параметры.
costsubtable	Задаёт процедуру, определяющую стоимость обработки данного блока, общую стоимость, стоимость обработки одного объекта и стоимость в единицу времени.

В дополнение к этим стандартным атрибутам и подобъектам многие блоки содержат собственные атрибуты, относящиеся только к данному блоку.

Настройка ReThink. Поведение блока (например, связанные с ним стоимость и время обработки) определяется атрибутами блока (например, duration-subtable) и реализуется с помощью процедур G2. В ReThink таким образом задается стандартное поведение блоков, но можно изменить это поведение, подставляя свои собственные процедуры G2. Например, можно задать свое распределение интервала времени между заявками, и поведение блока изменится соответствующим образом. Для запуска модели необходимо активизировать сценарий, проинициализировать таймер и запустить моделирование.

Динамическое создание определений объектов. При запуске модели на выполнение ReThink создает определения классов для этих объектов и помещает их на рабочее пространство. Классы, определяемые в модели ReThink, могут иметь широкий набор атрибутов, сложную структуру, находиться в различных отношениях с другими классами. Из построения модели в конечном итоге естественным образом вытекает создание все более сложных определений классов для объектов этой модели. Впоследствии эти классы могут послужить основой реализации информационных систем, используемых в поддержке ваших бизнес-процессов.

Стандартные атрибуты рабочих объектов. Как и атрибуты блоков, атрибуты рабочих объектов накапливают статистику, которую можно использовать для оценки эффективности процесса. Например, атрибут total-cost рабочего объекта вычисляет суммарную стоимость всех работ, выполненных над этим объектом. ReThink записывает стоимостные, временные и другие параметры и передает их вместе с объектом по мере его обработки.

Операции в блоке Задача. Каждый объект-операция имеет свое время начала и время окончания, и для каждой операции учитываются ее продолжительность и стоимость. Продолжительность и стоимость отдельных операций вносят свой вклад в общую продолжительность и стоимость для блока.

В любое время можно узнать, сколько объектов сейчас обрабатывает блок с помощью атрибутов блока, относящихся к операциям.

К операциям в блоке относятся следующие атрибуты:

Атрибут	Описание
total-starts	Число операций, начатых данным блоком.
total-stops	Число операций, завершённых данным блоком.
current-activities	Разность атрибутов total-starts и total-stops, т.е. число операций, выполняемых в настоящее время.
maximum-activities	Максимально допустимое число операций, выполняемых блоком одновременно.

Инструменты (instruments). Существуют два класса инструментов (см. рис. 9.3): установщики (feeds), устанавливающие значения, и зонды (probes), получающие значения от процесса.

Установщики (feeds) используются для присвоения значений в ходе моделирования. С их помощью можно присваивать значения множеству объектов процесса: блокам, рабочим объектам или путям. Установщик изображается пиктограммой в виде треугольника вершиной вниз, что означает «закачку» значений в модель.

Блоки оперируют объектами: создают их, устанавливают между ними связи, уничтожают их и т.д. В ходе моделирования, кроме того, происходит присвоение значений параметрам: сколько времени уходит на выполнение задачи, какова стоимость части процесса и т.п. Для выдачи этих значений можно использовать установщики.

В ReThink имеются следующие установщики:

Установщик	Описание
Timestamp (временная отметка)	Устанавливает временную отметку на объекте.
Accumulate (накапливание)	Накапливает значения. Например, с помощью установщика Accumulate можно накапливать затраты, связанные с производством объекта.
Count (счет)	Подсчитывает события. Например, установщик Count можно использовать для подсчета количества прохождений объекта через цикл обработки.
Change (присвоение атрибута)	Осуществляет присваивание значения константы любого типа. Например, с помощью установщика Change можно назначить имя клиента определенному заказу.

Зонды (probes) служат для получения значений из процесса. Зонд обозначается пиктограммой в виде треугольника вершиной вверх. Это означает получение значений из процесса. В ReThink имеются следующие зонды:

Зонд Delta Time (временной интервал)	Описание Измеряет временные интервалы. Зонд Delta Time вычисляет разницу между временной отметкой, созданной установщиком Timestamp (или одной из временных отметок, имеющихся у сущности по умолчанию), и текущим модельным временем. Зонд вычисляет и сохраняет это значение.
Sample Value (значение атрибута)	Извлекает значение атрибута из любого объекта модели. Например, можно извлекать значения атрибутов блоков, рабочих объектов, путей или ресурсов. Это простейшая разновидность зонда.
Moving Average (изменяющееся среднее)	Вычисляет изменяющееся среднее значение некоторого атрибута и стандартное отклонение от среднего.

ReThink накапливает историю значений, получаемых посредством зонда, а также историю значений, вычисляемых некоторыми зондами, в одном из атрибутов инструмента. В этом атрибуте определяется подобъект, являющийся объектом-параметром G2. Параметры хранят историю своих значений, что позволяет легко строить график этих значений (см. Приложение 1).

Как и в случае блоков, все определения классов и методы для инструментов определяются пользователем. Можно создавать собственные зонды и установщики. Например, можно создать зонд сигнала тревоги, который подает сигнал и высвечивает сообщение при достижении некоторого критического значения.

Для создания графика значений истории необходимо создать объект, называемый *передатчиком* (remote). Передатчик создается непосредственно из инструмента (instrument), а график – непосредственно из передатчика. На одном графике для сравнения можно строить несколько последовательностей значений. Как правило, все графики переносятся на подпространство сценария или на подпространство другого объекта.

Ресурсы. Важным элементом при моделировании процессов являются ресурсы, используемые при выполнении той или иной деятельности. Например, при моделировании процесса продажи для обработки заказов необходимо некоторое количество служащих, а для доставки грузов требуются грузовики.

Для ограничения модели ресурсы в ReThink могут использоваться следующими способами:

- Задача может требовать некоторое количество различных ресурсов. Например, для доставки грузов требуются грузовики и водители.
- Задача может иметь некоторое количество различных доступных ей ресурсов. Например, задача доставки грузов может иметь резерв доступных ей грузовиков и водителей. Чем больше доступный

ресурс, тем больший объем грузов может быть перевезен в заданное время. Чем меньше доступный ресурс, тем больше ограничивается задача доставки ресурсов.

- Задача может требовать часть ресурса, например половину компьютера.

- Некоторый ресурс может совместно использоваться различными задачами, например, служащий может выполнять работы для двух различных отделов.

Каждый ресурс, используемый в модели, можно охарактеризовать как фиксированной, так и переменной стоимостью, что позволяет вычислять стоимость процессов модели.

В процессе моделирования исходные материалы, ресурсы и промежуточные результаты трудовой деятельности перемещаются от блока к блоку в соответствии с указанным пользователем направлением материальных и информационных потоков. Каждый из элементов модели (блоки, ресурсы, предметы труда) обладает рядом стоимостных и временных атрибутов, к которым можно получить доступ непосредственно через таблицу атрибутов или путем использования инструментальных средств: для установки и для получения значений. Инструменты служат для связи контролируемых параметров с формами отчетов и графиков.

На палитре ресурсов, изображенной на рис. 9.3, представлены два типа ресурсов: грузовики и люди-водители (см. Приложение 2).

Палитра ресурсов содержит абстрактные буфера ресурсов, которые можно использовать для представления различного рода хранилищ: складов, БД и т.п. Из любого ресурса может быть создан менеджер ресурсов (отображаемый маленьким кружком), присоединение которого к любому из блоков задач указывает на необходимость использования определенного ресурса при выполнении данной задачи.

В процессе моделирования исходные материалы, ресурсы и промежуточные результаты трудовой деятельности перемещаются от блока к блоку в соответствии с указанным пользователем направлением материальных и информационных потоков. Каждый из элементов модели (блоки, ресурсы, предметы труда) обладает рядом стоимостных и временных атрибутов, к которым можно получить доступ непосредственно через таблицу атрибутов или путем использования инструментальных средств для установки и получения значений. Инструменты служат для связи контролируемых параметров с формами отчетов и графиков.

ЛИТЕРАТУРА

1. Попов Э. В. Бизнес-процесс “Реинжиниринг” и интеллектуальное моделирование компаний // Статические и динамические экспертные системы / Учеб. пособие. - М.: Финансы и статистика, 1996.

2. Попов Э.В., Шапот М.Д. **Реинжиниринг бизнес-процессов и информационные технологии** // Открытые системы. - 1996. - № 1.
3. Шапот М.Д. **Инструментальные средства поддержки реинжиниринга бизнес-процессов** // Материалы семинара "Динамические интеллектуальные системы в управлении и моделировании". - М.: ЦРДЗ, 1996.
4. Davenport T.H. **Business Innovation, Reengineering Work through Information Technology**. - Boston: Harvard Business School Press, 1993.
5. Flynn K. **Critical Success Factors for a Successful Business Reengineering Project** //CASE World Conference Proceedings. - Boston, 1993, October.
6. Hammer M. and Champy J. **Reengineering the Corporation: A Manifesto for Business Revolution**. - N-Y.: HarperCollins, 1993.
7. Harmon P. **Business Process Reengineering with Objects - Part 2** //Object-Oriented Strategies. - 1995. - V. 5. - № 1.
8. Jacobson I., Ericsson M., Jacobson A. **The Object Advantage: Business Process Reengineering with Object Technology** //ACM Press. - Addison-Wesley Publishing, 1995.
9. Kleinberg. **BPR Tools Categories - Multiple Choices Research Note** // July 7. 1995. GartnerGroup.
10. Salgame R.R., Becker S.G., Yu D.H. **Sparks: A Knowledge Based Process Modeling and Simulation System** // The National Business Process Reengineering Conference - 1995.

П1

G2 - ИНСТРУМЕНТАЛЬНЫЙ КОМПЛЕКС ДЛЯ СОЗДАНИЯ ДИНАМИЧЕСКИХ ИНТЕЛЛЕКТУАЛЬНЫХ СИСТЕМ В УПРАВЛЕНИИ И МОДЕЛИРОВАНИИ

История развития инструментальных средств (ИС) для создания динамических интеллектуальных систем реального времени началась в 1985 г., когда фирма Lisp Machine Inc. выпустила систему Picon для символьных ЭВМ Symbolics. Успех этого инструментального средства привел к тому, что группа ведущих разработчиков Picon в 1986 г. образовала частную фирму Gensym (США), которая, значительно развив идеи, заложенные в Picon, в 1988 г. вышла на рынок инструментальных средств под названием *G2 версия 1.0*. В настоящее время функционирует версия 4.2 и готовится к выпуску 5.0.

Основное назначение программных продуктов фирмы Gensym [4] - помочь предприятиям сохранять и использовать знания и опыт их наиболее талантливых и квалифицированных сотрудников в интеллектуальных системах реального времени, повышающих качество продукции, надежность и безопасность производства и снижающих производственные издержки. О том, как фирме Gensym удастся справиться с этой задачей, говорит хотя бы то, что сегодня ей принадлежат 50% мирового рынка интеллектуальных систем, используемых в системах управления [8].

Классы задач, для которых предназначена система G2:

- мониторинг в реальном масштабе времени;
- системы моделирования;
- системы обнаружения неисправностей;
- диагностика;
- составление расписаний и планирование;
- оптимизация;
- системы-советчики оператора;
- системы проектирования.

Немалую долю успеха фирме Gensym обеспечивают принципы, которых она придерживается в своих новых разработках:

- проблемно-предметная ориентация;
- следование стандартам;

- независимость от вычислительной платформы;
- совместимость снизу вверх с предыдущими версиями;
- обеспечение технологической основы для прикладных систем;
- комфортная среда разработки;
- поиск новых путей развития технологии;
- распределенная архитектура “клиент-сервер”;
- высокая производительность.

Основное достоинство оболочки экспертных систем G2 для российских пользователей - возможность применения ее как интегрирующего компонента, позволяющего за счет открытости интерфейсов и поддержки широкого спектра вычислительных платформ легко объединять уже существующие, разрозненные средства автоматизации в единую комплексную систему управления, охватывающую все аспекты производственной деятельности - от формирования портфеля заказов до управления технологическим процессом и отгрузки готовой продукции. Это особенно важно для отечественных предприятий, парк технических и программных средств которых формировался по большей части бессистемно, под влиянием резких колебаний в экономике. В связи с тем что экранные сообщения G2 русифицированы, российские пользователи общаются с системой G2 на русском языке.

Кроме системы G2, как базового средства разработки, фирма Gensym предлагает комплекс проблемно/предметно-ориентированных расширений для быстрой реализации сложных динамических систем на основе специализированных графических языков, включающих параметризуемые операторные блоки для представления элементов технологического процесса и типовых задач обработки информации. Набор инструментальных сред фирмы Gensym, сгруппированный по проблемной ориентации включает:

- интеллектуальное управление - G2 [6], G2 Diagnostic Assistant (GDA) [5], NeurOn-Line (NOL) [7], Statistical Process Control (SPC), BatchDesign_Kit;
- оперативное планирование - G2, G2 Scheduling Toolkit (GST), Dynamic Scheduling Packadge (DSP);
- реинжиниринг бизнес-процессов - G2, ReThink, BatchDesign_Kit;
- управление операциями и корпоративными сетями - G2, Fault Expert.

Несмотря на то что первая версия системы G2 появилась не так давно - в 1988 г. и ее даже в богатой Америке никто не назовет дешевой, G2 можно назвать бестселлером на рынке программных продуктов - на начало 1996 г. в мире было установлено более 6 000 ее копий. Фирма Gensym обслуживает более 30 отраслей - от аэрокосмических исследований до производства пищевых продуктов. Список пользователей G2 выглядит как справочник Who-Is-Who в мировой промышленности. Все 25 самых крупных промышленных миро-

вых корпораций используют G2. На базе G2 написано более 700 действующих приложений.

Успех инструментального комплекса G2 объясняется следующим: G2 - это динамическая система; G2 - это объектно-ориентированная интегрированная среда для разработки и сопровождения приложений реального времени, использующих базы знаний; G2 функционирует на большинстве существующих платформ (табл. П1.1). База знаний G2 сохраняется в обычном ASCII-файле, который однозначно интерпретируется на любой из поддерживаемых платформ. Перенос приложения не требует его перекомпиляции и заключается в переписывании файлов. Функциональные возможности и внешний вид приложения не претерпевают при этом никаких изменений.

Ниже будут рассмотрены основные компоненты G2: база знаний, машина вывода, подсистема моделирования и планировщик.

Т а б л и ц а П1.1

Платформы, на которых функционирует G2

Фирма - произво- дитель	Вычислительная система	Операционная среда
Digital Equipment	VAX 3xxx, 4xxx, 5xxx, 6xxx, 7xxx, 8xxx, DECstation 2xxx, 3xxx, 5xxx DEC Alpha AXP	VMS ULTRIX Open VMS, OSF/1, Windows NT
SUN Microsystems	SUN-4 SPARC 5, 10, 260, 280, 330, 370, LX	Sun OS, Solaris 2.x
Hewlett Packard	HP 9000/4xx, 7xx, 8xx	HP-UX
IBM	RISC 6000	AIX
Data General	AViiON	DG/UX
Silicon Graphics	IRIS	IRIX
Intel	Intel 486/Pentium	Windows NT, Windows-95, Windows 3/1
Motorola	Motorola 88000	UNIX
NEC	EWS 4800	EWS-UX/V

П1.1

База знаний

Все знания в системе G2 хранятся в двух типах файлов: базы знаний (БЗ) и библиотеки знаний (БиЗ). В файлах БЗ хранятся знания о приложениях (определения всех объектов, объекты, правила, процедуры и т.п.), в файлах БиЗ - общие знания, которые могут быть использованы более чем в одном приложении, например определение стандартных объектов. Файлы БЗ имеют расширение kb (knowledge base), а файлы БиЗ - расширение kl (knowledge libraries). Файлы БЗ могут путем замены расширения преобразоваться в БиЗ и обратно.

В целях обеспечения повторной используемости приложений в системе G2 реализовано средство, позволяющее объединять ранее созданные kb- и kl-файлы с текущим приложением. При этом G2 автоматически выявляет и выводит на дисплей конфликты в объединяемых знаниях.

Знания в G2 структурируются следующими способами: иерархия классов, иерархия модулей, иерархия рабочих пространств. Каждая из указанных иерархий может быть показана на дисплее, для чего используется возможность операции "Inspect".

П1.1.1

Сущности и иерархия классов

Класс служит основой представления знаний в системе G2. Понятие "класс в G2" базируется на объектно-ориентированной технологии (ООТ). Использование ООТ является главной тенденцией на текущем уровне развития искусственного интеллекта и вообще программирования.

В ООТ структуры данных представляются в виде классов объектов (определений объектов), имеющих определенные атрибуты. Классы наследуют атрибуты от суперклассов и передают свои атрибуты подклассам. Каждый класс (исключая корневой) может иметь конкретные экземпляры класса. В четвертой версии G2 введен механизм множественного наследования. Теперь в системе достаточно легко произвести новый класс, например саморегулирующихся насосов от классов контроллеров и насосов. В системе достаточно изяшно решена проблема конфликтов между именами атрибутов. Использование ООТ обеспечивает следующие преимущества:

- 1) уменьшает избыточность и упрощает определение классов, так как определяется не весь класс, а только его отличие от суперкласса;
- 2) позволяет применять общие правила, процедуры, формулы, что уменьшает их количество;

3) является естественным для человека способом описания сущностей.

Класс в G2 служит основой представления знаний. Все, что хранится в БЗ и чем оперирует система, является экземпляром того или иного класса. Все синтаксические конструкции G2 также представляют собой классы. Для сохранения общности даже базовые типы данных - символьные, числовые, булевские и истинностные значения нечеткой логики - представлены соответствующими классами. Описание класса (или экземпляра специального класса) включает в себя ссылку на суперкласс (is-a-иерархия) и перечень атрибутов, специфичных для данного класса.

Концептуально иерархия классов G2 берет свое начало от корневого класса, именуемого *item-of-value* (сущность или значение). Класс *item-of-value* сам по себе не может иметь экземпляров. А так как он является корнем всей иерархии классов, он определяет основное поведение всех классов G2. *Item-of-value* имеет два производных класса: *value* (хотя концептуально ветвь *value* представляется классом, в действительности это типы данных G2) и *item*. Каждый из этих классов имеет свои производные классы. Сущность (*item*) является корнем разветвленной иерархии классов. Наиболее важные ветви этой иерархии могут быть сгруппированы в небольшое число категорий. Они перечислены ниже (см. П1.1.3), в порядке их "видимости" для пользователя, начиная с наиболее "видимых".

П1.1.2

Иерархия модулей и рабочих пространств

G2-приложение - единый блок: оно структурируется с помощью модулей и рабочих пространств на легкоуправляемые части. Несмотря на то что функции модулей и рабочих пространств во многом схожи друг с другом, между ними есть существенные различия, отмеченные ниже.

G2-приложение может быть организовано в виде одной БЗ или в виде нескольких БЗ, называемых *модулями*. В последнем случае говорят, что приложение модулировано (структурировано на модули). Модули приложения организованы в древовидную иерархию с одним модулем верхнего уровня. Модули следующего уровня состоят из тех модулей, без которых не может работать модуль предыдущего уровня. Эти модули называют *непосредственно требуемыми модулями*.

Существуют два способа создания G2-приложения:

- 1) Разрабатывается одномодульное приложение, которое при необходимости разделяется на отдельные модули.

② Приложение изначально создается как состоящее из нескольких модулей. Некоторые из этих модулей разрабатываются впервые, а другие могут выбираться из библиотеки знаний.

Структурирование приложения на модули обеспечивает следующие преимущества:

- позволяет разрабатывать приложение одновременно нескольким группам разработчиков;
- упрощает разработку, отладку и тестирование;
- позволяет изменять модули независимо друг от друга;
- упрощает повторное использование знаний.

Рабочие пространства являются контейнерным классом, в котором размещаются другие классы и их экземпляры, например объекты, связи, правила, процедуры и т.д. Каждый модуль (база знаний) может содержать любое количество рабочих пространств. Рабочие пространства образуют одну или несколько древовидных иерархий с отношением *is-a-part-of* (является частью). С каждым модулем (базой знаний) ассоциируется одно или несколько рабочих пространств верхнего (нулевого) уровня, каждое из этих рабочих пространств является корнем соответствующей древовидной иерархии. В свою очередь, с каждым объектом (определением объекта или связи), расположенным в нулевом уровне, может быть ассоциировано рабочее пространство первого уровня, связанное с ним отношением “является частью”, и т.д.

Различие между модулями и рабочими пространствами состоит в следующем. Модули разделяют приложение на отдельные базы знаний, совместно используемые в различных приложениях. Динамические модули (аналог библиотек динамического связывания) могут подгружаться и вытесняться из оперативной памяти во время их выполнения программно и одновременно использоваться несколькими приложениями. Рабочие пространства играют свою роль при выполнении приложения: они содержат в себе (и в своих подпространствах) различные сущности и обеспечивают разбиение приложения на небольшие части, которые легче понимать и обрабатывать. Например, весь процесс разбивается на подпроцессы, и с каждым подпроцессом ассоциируется свое подпространство.

Рабочие пространства могут устанавливаться (вручную или действием в правиле-процедуре) в активное или неактивное состояние (т.е. сущности, находящиеся в этом пространстве и в его подпространствах, становятся невидимыми для механизма вывода). Механизм активации (деактивации) рабочих пространств используется, например, при наличии альтернативных групп правил, когда активной должна быть только одна из альтернативных групп.

Кроме того, рабочие пространства используются для задания ограничений, определяющих поведение приложения для различных категорий пользователей.

Структуры данных БЗ

Глобально сущности в БЗ G2 с точки зрения их использования могут быть разделены на структуры данных и исполняемые утверждения. Примерами первых являются объекты и их классы, связи (connection), отношения (relation), переменные, параметры, списки, массивы, рабочие пространства и т.п., примерами вторых - правила, процедуры, формулы, функции и т.п.

Опишем наиболее важные ветви иерархии "item".

Объект (object) и его подклассы. Объекты представляют объекты реального мира в приложении. Класс объектов определяет атрибуты, которые позволяют создать пиктограммы для объектов, определить их положение на схемах и *отростки связей (stubs)* для присоединения их к другим объектам.

Связь (connection). Этот класс предназначен для изображения путей между объектами. Можно создать подкласс связей для указания различных типов потоков, которые могут существовать между объектами на схеме. Например, объекты могут быть соединены водопроводными трубами и (или) проводами, передающими логические сигналы. Определив различные классы для этих связей, можно быть уверенным, что система G2 будет их различать и никогда не позволит "воде течь по электропроводам".

Рабочее пространство БЗ (Kb-workspace). Класс, определяющий независимый сегмент базы знаний, который может быть активирован или деактивирован. Рабочие пространства отображаются как отдельные, ограниченные рабочие области, в которые можно помещать объекты и объединять их в схемы. Связи создаются между рабочими пространствами с помощью *точек связи (connection posts)*. По сути, класс рабочих пространств является развитием концепции рабочей памяти в традиционных экспертных системах. Можно сказать, что рабочая память системы G2 строится на основе иерархии рабочих пространств. Иерархия рабочих пространств тесно связана с графическим представлением объектов. Рабочее пространство является контейнерным классом для экземпляров других классов. Каждый экземпляр объекта может обладать своим рабочим пространством, представляющим его внутреннюю структуру. Введение концепции рабочих пространств обеспечивает две важные функции системы G2: возможность претворять в жизнь рассуждения на разных уровнях абстракции и возможность продолжительной (теоретически - бесконечной) работы системы без необходимости "сборки мусора" в пределах отведенного объема оперативной памяти, что очень важно для систем управления непрерывными процессами.

Классы пользовательского интерфейса (user-interface). Эти классы определяют такие элементы пользовательского интерфейса, как меню, селективные кнопки (radio button), сообщения (message), шкалы, круговые шкалы и многое другое. Можно определять подклассы класса сообщений, например, для создания сообщений со специальным способом отображения. Из всех классов пользовательского интерфейса только для сообщений есть возможность создавать производные классы.

Классы описаний классов (class definition) определяют классы, экземпляры которых содержат созданные пользователем описания классов и служат шаблонами для создания экземпляров других классов. Эти классы порождены от класса описаний (definition). Описание имеет три определяемые пользователем подкласса - *описание объекта (object-definition)*, *описание связи (connection-definition)* и *описание сообщения (message-definition)*.

Классы языка G2 (G2 language): эти классы используются для определения различных элементов языка G2, таких, как правила, отношения, действия и процедуры. Нельзя создать собственные производные классы от этих классов.

С помощью G2 новые классы могут создаваться не только в процессе разработки, но и динамически, во время работы приложения. Во время выполнения приложения может быть создан, модифицирован или уничтожен экземпляр любого класса или целый класс. Это касается как объектов, так и правил, и процедур. В этом смысле G2 - "более объектно-ориентированная" система, чем даже C++. Описанная возможность является частью общих возможностей G2, дополняющих описания классов и позволяющих создавать новые сущности, включая рабочие пространства, правила, связи и процедуры. G2 обеспечивает операции *create by cloning* (создание клонированием) и *change the text of* (изменить текст), которые используются для клонирования похожего описания класса и последующего его редактирования. По умолчанию динамически созданные сущности являются *временными (transient)*, т. е. они существуют только на протяжении данного сеанса работы и не сохраняются в базе знаний. Однако описание класса должно быть *постоянной (permanent)* сущностью в момент создания экземпляра или производного класса. Операция *make permanent* (сделать постоянным) используется для преобразования временной сущности - описания класса - в постоянную.

Все классы G2 обладают по крайней мере одним общим свойством - их экземпляры имеют графическую форму представления. С помощью графических образов и классов связей можно строить схемы систем для любого уровня сложности. G2 позволяет "осуществлять рассуждения" на основе графических схем. Например, можно проверить состояние всех вентилей, соединенных с данной емкостью, или определить температуру объекта, ближайшего к указанному.

Объекты (классы) G2 подразделяются на встроенные в систему и вводимые пользователем. При разработке приложения, как правило, создаются подклассы пользовательских и встроенных классов, отражающие специфику данного приложения. Среди встроенных подклассов наибольший интерес представляет подкласс объектов, включающий подклассы переменных и параметров, и подкласс связей (connection) и отношений (relation).

Объекты

Объекты в базе знаний представляют собой отображения элементов реального мира, которые будут использоваться при решении поставленной перед G2 задачи. Выделяют постоянные и временные объекты. *Постоянные объекты* заносятся в БЗ разработчиком G2 в процессе диалога с системой, в то время как *временные объекты* создаются после выполнения специальных команд в правилах и процедурах. Временные объекты могут существовать в БЗ только в процессе работы G2. С каждым объектом ассоциируется таблица атрибутов, в которую заносятся существенные для решаемой задачи свойства объекта. Элемент данной таблицы представляет собой пару "атрибут - значение".

Объекты могут иметь графические образы, отображаемые на экране дисплея, - *пиктограммы (иконки)*. На пиктограммах разработчиком могут быть выделены отдельные участки. Цвет таких участков может изменяться в результате выполнения специальных команд в правилах или процедурах. Таким способом обеспечивается высокая наглядность информации, предоставляемой лицу, работающему с G2.

Поскольку реальные приложения могут содержать большое количество объектов, G2 обеспечивает возможность объединения множества объектов со схожими свойствами в классы. Классы объектов составляют иерархию, в которой определяется отношение "родительский класс - подкласс". Объекты подклассов могут наследовать атрибуты и пиктограммы родительских классов.

Иерархическая упорядоченность классов значительно упрощает задачу определения новых классов в приложении. Например, атрибуты, характеризующие объекты различных классов, могут быть однократно определены в одном классе, являющемся общим родительским классом для них. Такие атрибуты будут автоматически наследоваться объектами, принадлежащими к подклассам, что снимает необходимость их повторного определения. Другое важное достоинство введения классов объектов - возможность составления правил, относящихся ко всем объектам, принадлежащим к некоторому классу (общие правила). Задача разработчика значительно упрощается благодаря тому, что он может применять общие правила, относящиеся к различным классам объектов. При этом результирующая БЗ будет иметь меньший объем по сравнению с БЗ, в которой не могут применяться общие правила.

Особая роль в G2 отводится переменным. В отличие от статических систем, переменные в G2 делятся на три вида: параметры, собственно переменные и простые атрибуты. *Параметры* получают значения в результате работы механизма вывода или выполнения какой-либо процедуры. *Переменные* представляют измеряемые характеристики объектов реального мира и поэтому имеют специфические черты: время жизни значения и источник данных. *Время жизни значения переменной* определяет промежуток времени, в течение которого данное значение актуально, по истечении этого промежутка переменная считается не имеющей значения. В отличие от переменных параметры всегда имеют значение, так как их значения либо заданы в качестве начальных значений, либо перевычислены механизмом вывода G2.

Поскольку системе может потребоваться текущее значение переменных, для каждой из них должен быть определен *источник данных (сервер данных)*. Источником данных для переменной могут служить: машина вывода, подсистема имитационного моделирования или внешний по отношению к G2 источник данных. С переменными могут быть ассоциированы формулы имитационного моделирования, в результате применения которых система также может получать значения переменных. Для параметров указанный механизм получения значений из источника данных не используется; они получают новые значения после выполнения специальных операторов в заключениях правил или процедур.

При ссылке в правиле или процедуре как для переменных, так и для параметров допустимо использование следующих выражений, отражающих динамику их значений:

- текущее значение;
- значение в заданный момент времени;
- среднее значение за интервал времени;
- интеграл по интервалу времени;
- интерполяция значения в заданный момент времени;
- максимальное (минимальное) значение за интервал времени;
- количество собранных значений за интервал времени;
- скорость изменения значений в течение интервала времени;
- стандартное отклонение в течение интервала времени.

Очевидно, что далеко не для всех используемых в приложении значений нужно применять такой мощный инструментальный, поэтому в целях повышения эффективности функционирования системы в этих случаях используют *простые атрибуты*.

Связи и отношения

В G2 предусмотрены два вида взаимосвязей между объектами: связи и отношения. Под *связями* понимается взаимосвязь между двумя сущностями, задаваемая разработчиком приложения и имеющая графическое представление. В реальном физическом окружении.

описываемом в G2, связи может соответствовать физическая связь между сущностями, такая, как электрическое соединение или трубопровод. В G2 разработчик может задавать классы связей, ссылаться на объекты посредством указания связей, в которых они участвуют, а также делать заключения на основании наличия или отсутствия связей. Под *отношением* понимается поименованная взаимосвязь между двумя сущностями. G2 предоставляет возможность разработчику задавать различные типы отношений. На основании наличия или отсутствия отношения между объектами делаются выводы. Основные отличия связей и отношений:

- связи задаются разработчиком в процессе создания экспертной системы, в то время как отношения устанавливаются динамически после выполнения специальных операторов в правилах или процедурах;
- связи имеют графическое представление, в то время как отношения не отображаются на экране дисплея;
- отношения в отличие от связей нецелесообразно сохранять в качестве постоянной части БЗ.

Исполняемые утверждения БЗ

Основу исполняемых утверждений БЗ составляют правила и процедуры. Кроме того, есть формулы, функции, действия и т.п. Правила в G2 имеют традиционный вид: условие (антецедент) и заключение (консеквент). Кроме if-правила - условие ("if <логическое выражение>") и заключение ("then <действия>") используются еще четыре типа правил: unconditionally, when и whenever и initially.

Способы применения каждого правила определяются его синтаксисом:

```

<правило> ::= {[<префикс for>]
                {<правило if> | <правило unconditionally>
                | <правило when> | <правило whenever>}
                | <правило initially>}}
<префикс for> ::= for {any | the } <item> ...
<правило if> ::= if <логическое выражение> then <список действий>
<правило unconditionally> ::= unconditionally <список действий>
<правило when> ::= when <логическое выражение>
                    then <список действий>
<правило whenever> ::= whenever <описание события>
                    [or <описание события>]
                    [and when <логическое выражение>]
                    then <список действий>
<правило initially> ::= initially
                    [if <логическое выражение> then] <список действий>

```

Каждое из типов правил может быть как *общим*, т.е. относящимся ко всему классу, так и *специфическим*, относящимся к конкретным экземплярам класса. Возможность представлять знания в виде общих, а не только конкретных правил обеспечивает следующие преимущества:

- минимизацию избыточности БЗ;
- упрощение наполнения БЗ и ее сопровождения;
- минимизацию ошибок при отладке БЗ;
- повторную используемость знаний (так как общие правила записываются в библиотеке G2 и могут использоваться в подобных приложениях).

Несмотря на то что правила обеспечивают достаточную гибкость для описания реакций системы на изменения окружающего мира, в некоторых случаях, когда необходимо выполнить жесткую последовательность действий (например, запуск или остановку комплекса оборудования), более предпочтительным признан процедурный подход. Язык программирования, используемый в G2 для представления процедурных знаний, - достаточно близкий "родственник" Паскаля. Кроме стандартных управляющих конструкций язык расширен элементами, учитывающими работу процедуры в реальном режиме времени: ожидание наступления событий, разрешение другим задачам прерывать выполнение данной процедуры, директивы, задающие последовательное или параллельное выполнение операторов. Еще одна интересная особенность языка - *итераторы*, позволяющие организовать цикл над множеством экземпляров класса. Благодаря перечисленным свойствам языка в системе можно одновременно выполнять множество различных процедур или множество копий одной и той же процедуры для множества различных объектов.

П1.2

Машина вывода, планировщик и подсистема моделирования

П1.2.1

Машина вывода

Одним из основных компонентов G2 является машина вывода, выполняющая рассуждения на основании:

- знаний, содержащихся в базе знаний;
- данных, поступающих от подсистемы имитационного моделирования;
- данных, поступающих от внешних источников (контрольно-измерительной аппаратуры, СУБД и т. п.).

Правила возбуждаются машиной вывода. При этом проверяется истинность условия, находящегося в антецеденте правила. Если оно истинно, то машина вывода выполняет действия, находящиеся в кон-

секвенте. При проверке условия правила машине вывода необходимо найти значения всех переменных и параметров, содержащихся в них. Параметр в любой момент времени имеет определенное значение, в то время как значение переменной может отсутствовать, поскольку для переменной определено время жизни. Если переменная не имеет значения, система может получить ее значение из следующих источников:

- сервер данных, получающий данные от внешних источников (контрольно-измерительная аппаратура, СУБД, другая ЭС и т. п.);
- оператор;
- подсистема моделирования внешнего окружения;
- правило, определяющее искомое значение переменной;
- формула, приписанная к переменной (или классу переменных), значение которой требуется системе.

После того как все требуемые значения получены, система определяет, истинно ли условие рассматриваемого правила. Если условие правила истинно, система выполняет действия, находящиеся в консеквенте данного правила. В набор действий, которые используются в консеквентах правил, входят следующие действия:

- присвоение значения простому атрибуту, параметру или переменной;
- посылка управляющей информации внешнему объекту;
- запуск процедуры;
- создание экземпляра объекта;
- удаление экземпляра объекта;
- порождение и удаление задач (подзадач);
- изменение положения или цвета пиктограмм на экране дисплея;
- управление способом отображения рабочих пространств (положение на экране, масштаб и т. д.);
- вывод сообщений для оператора системы;
- активизация всех правил, ассоциированных с заданным объектом;
- остановка системы и т.д.

В связи с тем что G2 ориентирована на динамические приложения, работающие в реальном масштабе времени, машина вывода должна иметь средства для сокращения перебора, реакции на непредвиденные события и т.п. Главным недостатком традиционно используемого в статических ЭС прямого и обратного вывода является непредсказуемость затрат времени на их выполнение. Для динамических систем прямой и обратный вывод с полным перебором возможных к применению правил - непозволительная роскошь. Следующая образная интерпретация позволяет понять недостаток традиционных методов построения цепочек логического вывода и необходимость выхода на метауровень (см. ниже *focus* и *invoke*) в дина-

мических системах. Представьте себе, что вы пришли в библиотеку и хотите установить некоторый факт. Следуя алгоритму прямого вывода, вы должны начать читать все книги подряд в алфавитном порядке, пока не наткнетесь на нужный факт. Когда факт установлен и ваше знание о действительности изменилось, следуя тому же алгоритму, вы должны перечитать все книги, даже уже прочитанные!

Особенностью машины вывода G2 является богатый набор способов возбуждения правил. Правило в системе G2 может возбуждаться одним из девяти случаев. Особый интерес представляют пять способов возбуждения правил при возникновении некоторого события. Для реализации этих методов в G2 введен специальный тип правил, начинающихся с ключевого слова *whenever* (как только). Правила *whenever* возбуждаются в первую очередь и обладают наибольшим приоритетом, что гарантирует своевременную реакцию системы на изменения в окружающем мире. Правила этого типа не используются (по умолчанию) ни в прямом, ни в обратном выводе, они являются метаправилами и реагируют на события (перемещение объекта, установление (устранение) отношения, получение (неполучение) значения).

П1.2.2

Планировщик

В связи с тем что G2-приложение управляет множеством одновременно исполняемых задач, необходим Планировщик. Хотя пользователь никогда не взаимодействует с ним, Планировщик контролирует как всю активность, видимую пользователем, так и активность фоновых задач. Планировщик определяет порядок обработки задач, взаимодействует с источниками данных и пользователями, запускает процессы и осуществляет коммуникацию с другими процессами в сети (рис. П1.1).

Планировщик G2 циклически выполняет следующую последовательность шагов.

Шаг 1. Проверка наступления начала цикла: если это начало цикла, то Планировщик начинает цикл и переходит к следующему шагу.

Шаг 2. Планирование ожидающих задач: Планировщик формирует список задач, которые будут выполняться в данном цикле. Этот список называется очередью текущих задач.

Шаг 3. Обслуживание источников данных: Планировщик обменивается с каждым источником данных. Каждому из источников данных отводится не более 0,1 с на выполнение этой операции. Для источников, не закончивших обмен за выделенное время, планируются задачи для окончания передачи данных.

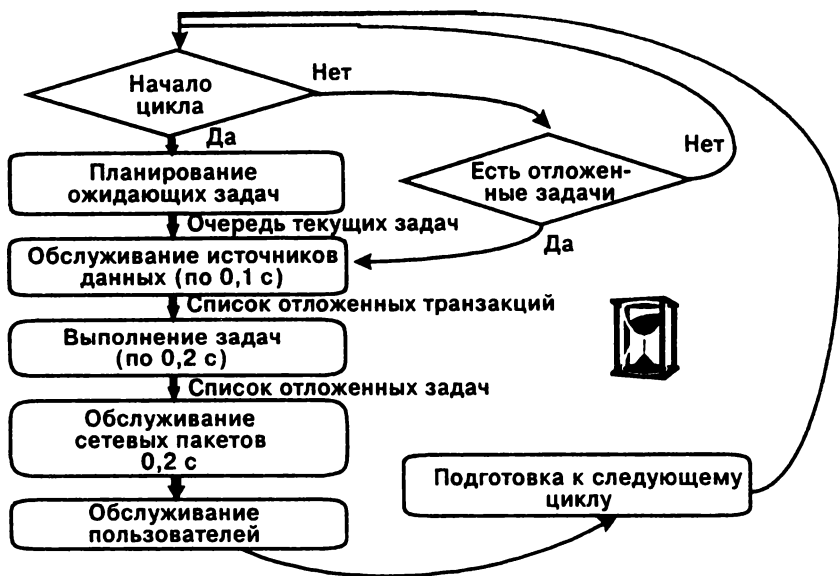


Рис.П1.1. Схема работы Планировщика G2

Шаг 4. Выполнение задач: Планировщик анализирует очередь текущих задач и пытается их выполнить. Любая из задач, не закончившаяся в течение 0,2 с, откладывается для выполнения в конце данного цикла или в следующем цикле.

Шаг 5. Обслуживание сетевых пакетов: Планировщик посылает и получает сообщения через сеть. На это выделяется до 0,2 с.

Шаг 6. Обслуживание пользователей: Планировщик принимает и передает данные для всех пользователей, работающих в данном сеансе G2. Это включает и пользователей Telewindows.

Шаг 7. Подготовка к следующему циклу: Планировщик проверяет, осталась ли активность в рамках данного цикла. Если осталась (получение данных, завершение отложенных задач и т. д.), он возвращается к шагу 1 и проверяет, не наступило ли время нового цикла. Если нет, он переходит к шагу 3 для завершения всех отложенных действий. Если да, Планировщик переходит на следующий цикл. Если отложенных задач не осталось и время нового цикла не наступило, Планировщик “засыпает” на 40 мс и после этого переходит к шагу 1.

Подсистема моделирования

Один из источников данных для G2 - подсистема моделирования внешнего окружения. Она используется для моделирования реальных объектов и устройств, с которыми работает ЭС. В подсистеме моделирования предусмотрены:

- средства для вычисления алгебраических уравнений, разностных уравнений и дифференциальных уравнений первого порядка;
- возможность задания формул как для отдельных переменных, так и для классов переменных или параметров;
- возможность режима разделения времени, при котором подсистема моделирования работает параллельно с остальными подсистемами G2. За счет этого осуществляется вычисление моделируемых значений в процессе работы механизма вывода G2.

Подсистема моделирования G2 является достаточно автономной, но важной частью системы. На различных этапах жизненного цикла приложения она служит различным целям. Во время разработки подсистема моделирования используется вместо объектов реального мира для имитации показаний датчиков. Очевидно, что отладка на реальных объектах может оказаться слишком дорогой, а иногда (например, при разработке системы управления атомной станцией) и небезопасной.

На этапе эксплуатации приложения подсистема моделирования обеспечивает:

- верификацию показаний датчиков во время исполнения приложения;
- подстановку модельных значений переменных при невозможности получения реальных (выход из строя датчика или получение ответа на запрос в течение длительного времени).

Играя роль самостоятельного агента знаний, подсистема моделирования повышает жизнеспособность и надежность приложений на базе G2. В системе G2 для описания внешнего мира подсистема моделирования использует уравнения трех видов: алгебраические, разностные и дифференциальные (первого порядка).

G2 выделяет три типа переменных, которые могут получать свои значения от подсистемы моделирования: *непрерывные, дискретные и зависимые*. Значения двух первых типов переменных являются функциями их предыдущих значений, вследствие чего для них должны быть заданы некоторые начальные значения. Значения зависимых переменных являются функциями только текущих значений других вычисляемых переменных. Эта категория переменных явно не объявляется; их значения получаются из уравнений моделирования для соответствующей переменной.

Приведем примеры алгебраических, разностных и дифференциальных уравнений, которые могут использоваться подсистемой моделирования G2 для вычисления объема некоторого объекта.

Если идентификатором данного объекта является бак_1, то алгебраической формулой является выражение:

объем бак_1 = уровень бак_1 * площадь бак_1 ,
где уровень и площадь - атрибуты объекта бак_1;
* - символ операции умножения.

Соответствующее конечно-разностное уравнение для объема бак_1 может иметь вид:

следующее значение объем бак_1 = объем бак_1 + 3
с начальным значением 5000.

Дифференциальное уравнение для этой же переменной (объем бак_1) выглядит так:

d/dt (объем бак_1) = (приток на входе бак_1 - отток на выходе бак_1) с
начальным значением 5000.

Подсистема моделирования, как правило, обеспечивает возможность задания формул моделирования не только для отдельных переменных и параметров, но и для их классов. За счет этого становится возможным однократное описание поведения всех экземпляров класса. Подсистема позволяет иметь также различные приращения времени при вычислении разных переменных.

Например, общая формула моделирования для объема любого водяного бака выглядит так:

d/dt (объем любой водяной_бак) = (приток на входе любой водяной_бак - отток на выходе любой водяной_бак) с начальным значением 5000.

Каждый параметр, значения которого задаются подсистемой моделирования, имеет атрибут алгоритм интегрирования, определяющий метод численного приближенного решения обыкновенных дифференциальных уравнений. В системе G2 для этих целей используют методы Эйлера и Рунге - Кутты. Выбор среди этих двух методов зависит от приложения.

Метод Рунге - Кутта дает более точные результаты, но требует в четыре раза больших затрат времени, чем метод Эйлера. Чтобы достигнуть той же точности, как в методе Рунге - Кутта, при использовании метода Эйлера необходимо значительно уменьшить время приращения. Таким образом, для приложений, которые требуют высокой точности и не предъявляют особых требований к скорости, часто эффективнее бывает использовать метод Рунге - Кутта. С другой стороны, для приложений, в которых существенна скорость, а не точность, целесообразнее применять метод Эйлера.

Устойчивость (сходимость) метода не является фактором, на основании которого следует выбирать тот или иной метод, так как и метод Эйлера, и метод Рунге - Кутта имеют один и тот же предел

устойчивости относительно размера шага приращения. Другими словами, максимальное время приращения, вне пределов которого вычисления становятся неустойчивыми, - одно и то же для обоих методов. Кратко поясним суть методов Эйлера и Рунге - Кутты [2].

Рассмотрим дифференциальное уравнение 1-го порядка $x = f(t, x)$ с начальным условием $x_0 = x(t_0)$.

Выбрав достаточно малый шаг приращения h , построим систему равноотстоящих точек $t_i = t_0 + ih$ ($i = 0, 1, 2, \dots$). Тогда в соответствии с методом Эйлера приближенные численные значения $x(t_i) \approx x_i$ вычисляются по формулам:

$$x_{i+1} = x_i + hf(t_i, x_i), \quad (i = 0, 1, 2, \dots).$$

А по методу Рунге - Кутты вычисление приближенного значения x_{i+1} в следующей точке $t_{i+1} = t_i + h$ производится по формулам:

$$\begin{aligned} x_{i+1} &= x_i + \Delta x_i, \\ \Delta x_i &= 1/6 (K_1 + 2K_2 + 2K_3 + K_4), \end{aligned}$$

где $K_1 = hf(t_i, x_i);$
 $K_2 = hf(t_i + h/2, x_i + K_1/2);$
 $K_3 = hf(t_i + h/2, x_i + K_2/2);$
 $K_4 = hf(t_i + h, x_i + K_3).$

Кроме внутренней подсистемы моделирования, система G2 имеет средства для подключения внешних, как правило, более мощных систем имитационного моделирования. В этом случае G2 может получать вычисляемые величины и устанавливать значения величин во внешних системах моделирования, а также выполнять часть процесса моделирования в самой G2 при выполнении другой части вне данной программы.

При работе с внешней подсистемой G2 обычно вызывает внешнюю систему имитационного моделирования для работы в качестве части подсистемы имитационного моделирования G2, так что другие компоненты G2, такие, например, как механизм вывода и интерфейс конечного пользователя, рассматривают ее в качестве единственного сервера данных. Например, механизм вывода может устанавливать значения во внешней системе имитационного моделирования так же, как он устанавливает значения в собственной подсистеме имитационного моделирования.

Такой подход реализован, в частности, при взаимодействии системы G2 фирмы Gensym с системой моделирования, реализованной в рамках комплекса Graphical Model Builder фирмы ABB Simcon, специализирующейся на разработках динамических моделей для нефтехимических предприятий.

П1.3

Среда разработчика в системе G2

Ранее мы охарактеризовали G2 как среду разработки и сопровождения приложений. Что же делает систему G2 именно средой, а не просто оболочкой для выполнения приложений реального времени, какой является, например, система RTworks фирмы Talarian Corporation? Средой являются средства разработчика, так как они автоматизируют взаимодействие с приложением на всех стадиях его существования: разработки, использования и сопровождения. Среда разработчика включает в себя прежде всего:

- естественно-языковой текстовый редактор, управляемый процедурой грамматического разбора;
- интерфейс с пользователем;
- средства инспекции и отладки;
- систему регистрации версий.

П1.3.1

Естественно-языковой текстовый редактор

Разработчик G2 предоставляет информацию о разрабатываемом приложении на ограниченном английском языке, и разработчику предоставлена возможность ссылаться на любую сущность в БЗ многими способами. Например, разработчик может использовать конструкции ЕЯ для того, чтобы сослаться на сущность следующими способами:

- 1) по имени: *pump-12* (насос-12);
- 2) с помощью префикса *for* (для) и слова *any* (любой), за которыми следует имя класса. Таким способом обеспечивается ссылка не на одну сущность, а на группу сущностей:

for any pump (для любого насоса) ...;

в результате можно записывать общие утверждения, например, такие:

for any tank
if the tank is empty
then inform the operator that "The tank ... is empty"

(для любой цистерны,
если эта цистерна пуста,
то информировать оператора, что "цистерна ... пуста");

- 3) как на одну из сущностей класса объектов, связанных с другим объектом:

the tank connected to the pump
(цистерна, связанная с насосом);

4) как на объект, ближайший на графической модели к некоторому объекту:

the pump nearest to the tank
(насос, ближайший к цистерне).

Несмотря на сложность и богатство синтаксических конструкций G2 для описания знаний, их применение упрощается за счет естественно-языкового подхода. Примером использования естественного языка в G2 для формирования базы знаний может служить следующее типичное правило:

If the altitude of any aircraft < the safe-flying-altitude of the aircraft then inform the operator that "Pull up. You are flying too low. Your altitude is [the altitude of the aircraft]".

Упрощение взаимодействия разработчика с системой достигается и за счет оригинального подхода, реализованного в текстовом редакторе. Интерактивный текстовый редактор G2 позволяет редактировать тексты утверждений, правил, функций, процедур и т. д. Он работает в специальном окне редактирования, которое появляется, как только инициируется создание нового утверждения или редактирование существующего, выбирается любой участок текста, добавляется или редактируется любой другой текст, включая текст, представляющий значение атрибута экземпляра класса. Процесс редактирования все время направляется процедурой грамматического разбора, что гарантирует введение только синтаксически правильных конструкций языка. В окне редактирования появляется динамически изменяемая подсказка, указывающая, какие языковые конструкции можно вводить, начиная с текущей позиции курсора. Есть возможность набирать вводимый текст на клавиатуре или выбирать подходящие шаблоны из подсказки. Кроме того, можно использовать клавиатурные команды или контекстно-зависимое меню операций редактирования, что упрощает редактирование. Например, правило, приведенное выше (исключая текст, заключенный в кавычки), может быть введено с помощью 16 нажатий клавиши мышки и введения с клавиатуры трех букв A и одной буквы S.

П1.3.2

Интерфейс с пользователем

Так же, как нет необходимости заучивать наизусть грамматические конструкции языка для написания правил и процедур G2, не нужно изучать и язык графических примитивов для построения пиктограмм объектов. Редактор пиктограмм (рис. П1.2) позволяет создавать пиктограммы графическими средствами и автоматически преобразовывать их в текстовые описания. В результате можно увидеть, как будет выглядеть пиктограмма (икона) и изменять ее.

Возможности графического интерфейса с пользователем настолько богаты, что даже их беглому описанию можно посвятить отдельную главу. Кроме того, они дополнены и поддержаны дизайнером диалогов (graphic user interface dialog editor - GUIDE), позволяющим разработчику выбирать стиль создаваемого интерфейса - MS Windows или Motif (XWindows). Ниже приводится краткий перечень основных возможностей графического интерфейса:

- использование растровой графики наряду с векторной для пиктограмм объектов и фоновых изображений;
- различные типы графиков и удобный интерфейс для их конфигурирования;
- возможность выбора между “слоеной” (перекрывающейся) или “полупрозрачной (XOR) отрисовкой” элементов интерфейса;
- произвольное масштабирование графических элементов;
- разнообразные векторные шрифты;
- поддержка формата Enhanced PostScript при сохранении изображения в файле для последующей печати;
- разнообразные способы работы с меню, текстовыми и графическими редакторами;
- планируемая на основе задаваемых пользователем приоритетов перерисовка отдельных участков экрана, гарантирующая первоочередное обновление наиболее важной информации на графиках, в диаграммах, таблицах и т. п.;
- разнообразные функции обработки клавиатуры и манипулятора “мышь”, позволяющие использовать специализированные интерфейсные решения при организации рабочего места оператора;
- поддержка стандарта ISO 8859-5 в отношении представления символов кириллицы независимо от операционной среды. Эта особенность открывает российским разработчикам возможность употребления русских имен в названиях классов, атрибутов и т. д.

Конечный пользователь может взаимодействовать с G2 различными способами. Так, в системе G2 разработчик создает разнообразные меню, которые дают возможность конечному пользователю:

- показать (скрыть) рабочее пространство;
- двигать или вращать сущность;
- изменять цвета икон и связей;
- выдавать сообщения и т.д.

В добавление к этим средствам взаимодействия G2 предоставляет конечному пользователю следующие средства:

- изображения (displays);
- управляющего воздействия на G2 (end-user controls);
- сообщения (messages);
- управления доступом (access control);
- создания опций меню (user menu choices);
- перевода опций меню.

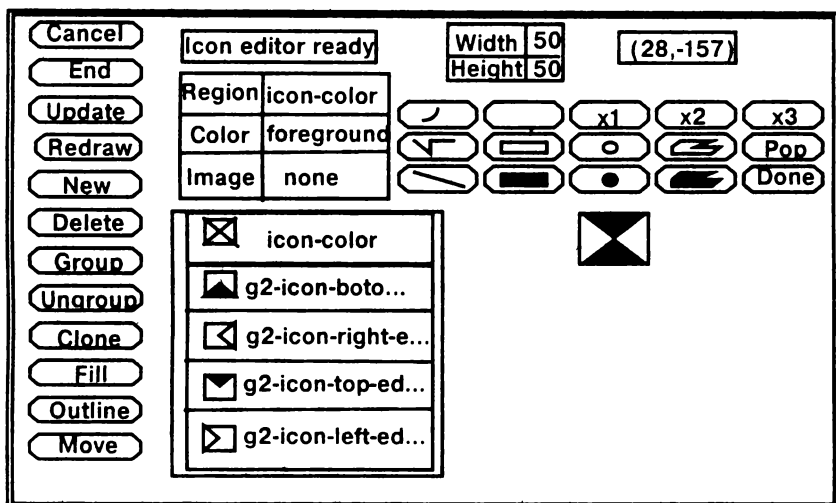


Рис.П1.2. Редактор пиктограмм G2

Изображения

Изображения используются для того, чтобы пользователь мог увидеть значение переменных и выражений. Существуют следующие варианты, реализующие возможность изображения (display):

Readout-table (отсчет). Показывает (в горизонтальном прямоугольнике) слева имя переменной (параметра) или вид выражения, а справа - значение.

Dial (циферблат). Изображает арифметическое значение в виде точки на круговой шкале. При движении по часовой стрелке значение увеличивается.

Meter (измеритель). Изображает в виде вертикальной шкалы с указателем значение арифметического выражения, переменной (параметра).

Graph (график). Изображает в двух измерениях изменение одного или более выражений (в частном случае параметра или переменной) во времени (рис. П1.3).

Chart (диаграмма). Отображает на графике (с двумя ортогональными осями) соотношение одной последовательности данных с некоторой другой последовательностью.

Freeform-table (таблица). Изображает таблицу из рядов и строк (подобна электронной таблице). Значение каждой ячейки в таблице может вычисляться G2.

Кроме того, система G2 позволяет реализовать возможность display динамически (т.е. в процессе работы механизма вывода) с помощью действия create (создать).

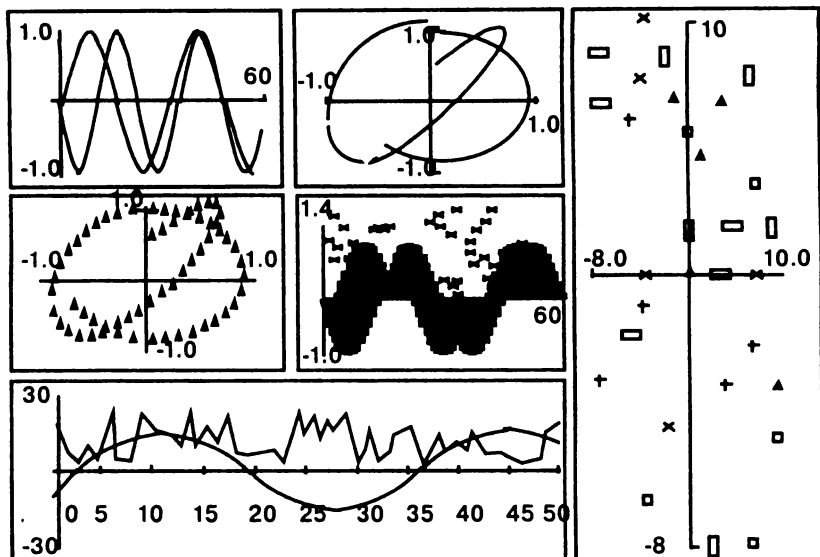


Рис. П1.3. Виды динамических графиков в системе G2

Управляющие воздействия

Управляющие воздействия (end-user controls) - это средства, с помощью которых конечный пользователь может взаимодействовать с приложением. Существуют следующие виды управляющих воздействий:

Action button (кнопка действия). Изображается в виде прямоугольника с закругленными углами. Нажатие разработчиком на кнопку действия приводит к выполнению связанных с ней действий, например, таких, как: change (изменить), conclude (закljučить), create (создать), delete (устранить), halt (остановить), hide (скрыть), inform (информировать), move (двигать), rotate (вращать), transfer (перенести) и т.п.

Radio button (радиокнопка). Изображается в виде группы маленьких круговых иконок, соответствующих взаимоисключающему множеству символов, чисел, логических или текстовых значений. При нажатии на одну из иконок соответствующее ей значение присваивается переменной или параметру. Например, пользователь может использовать группу radio buttons, помеченную red (красный), black (черный), white (белый), для назначения цвета некоторой символической переменной.

Check boxes (кнопка проверки). Изображается в виде маленького квадрата вместе со связанным с ним значением переменной или параметра (символическим, количественным, логическим, текстовым).

Check boxes проверяет, является ли указанное значение активным. Если значение активно (on), то в квадрате Check boxes появляется крест, если значение неактивно (off, т.е. какое-то другое значение), то в квадрате нет ничего; если значение не известно, то в квадрате стоит знак вопроса (?).

Slider (указатель). Изображается в виде горизонтальной шкалы с указателем значения для начала, конца и указателя. Перемещая указатель вдоль горизонтальной шкалы, пользователь может вводить числовое значение переменной или параметра.

Type-in box (ввод). Изображается в виде прямоугольника с горизонтальной длинной стороной. Значение (числовое или текстовое) переменной или параметра, связанное с Type-in box, может быть введено в прямоугольник пользователем с клавиатуры. Если какой-либо другой источник (не Type-in box) изменит значение переменной (параметра), то Type-in box отобразит это изменение.

Сообщения

Сообщения (messages) - это класс сущностей, которые содержат в себе текст. Сообщения являются средством, позволяющим G2 информировать пользователя о событиях. Например, в результате исполнения утверждения inform (информировать) G2 создает некоторое сообщение и размещает его на доске сообщений (message-board); при обнаружении некоторой ошибки G2 выдает сообщение в журнал оператора (the operator logbook).

Сообщения, которые G2 создает в результате выполнения действия inform, или сообщения как реакция на ошибку и т.п., являются примерами встроенного класса сущностей, называемого message (сообщение). Разработчик может создать подкласс класса message, который будет обладать уникальными характеристиками и атрибутами. Например, класс сообщений, называемый user-warning-message (предупреждающие пользовательские сообщения), может использовать текст с очень крупным шрифтом, размещенным на фоне заданного цвета. Сообщения стандартно выдаются на одно из двух рабочих пространств: на доску сообщений (message board) или в журнал оператора (logbook). Обычно на доску сообщений выдаются сообщения для пользователя, вызываемые действием inform. Вид доски сообщений управляется с помощью атрибута message-board-parameters в системной таблице (system tables). Сообщения об ошибках, системных условиях и предупреждения выдаются в журнал оператора. Вид журнала управляется через атрибут logbook-parameters в system tables.

Управление доступом

С помощью средств управления доступом (access control) разработчик может влиять на то, что конечный пользователь видит и может делать с БЗ. Например, разработчик может управлять доступом следующим образом:

- ограничить (restrict) строки меню, которые видит пользователь в каждом меню;

- ограничить пользователей в возможности передвигать сущности, устанавливать связи и т.п.;

- определить перечень атрибутов, которые может видеть пользователь в таблицах атрибутов, в рабочих пространствах, в объектах и т.д.;

- разрешить пользователю видеть атрибуты сущностей, но не редактировать их и т.п.;

- обеспечить автоматическое выполнение некоторого действия, например показать рабочее пространство объекта, когда пользователь указывает на сущность.

Ограничения (restrictions), которые назначает разработчик, могут действовать:

- на все сущности в базе знаний;

- на определенные классы сущностей;

- на сущности определенного рабочего пространства;

- на частные сущности.

Разработчик управляет доступом, указывая тип пользователя (mode), который работает с приложением: оператор, разработчик, администратор и т.п. Разработчик может расширить список типов пользователей по своему усмотрению. Тип пользователя “администратор” встроен в G2, этот тип пользователя свободен от всех ограничений, он может видеть и делать все, на что способна G2.

Создание опций меню

Разработчик может определить новые опции (строки) меню сверх тех, которые используются стандартно. Когда пользователь выбирает опцию меню (user menu choice - umc) для того, чтобы внести новую строку в меню, соответствующую частному классу сущностей, система G2 выполняет определенные действия. Дополнительная строка в меню появляется для примеров соответствующих сущностей во время выполнения базы знаний и при соблюдении условий, указанных в опции umc.

Например, предположим, что база знаний содержит класс объектов, называемый “переключатель”, и этот класс имеет атрибут “состояние” со значением “включено” (on) или “выключено” (off). Разработчик может с помощью umc добавить строку к меню “переключатель”, которая в состоянии переключателя on будет содержать в меню строку off, и наоборот.

Перевод опций меню

G2 позволяет перевести имена любых опций меню с английского на другой язык. Это обозначает, что разработчик G2 может заменить некоторые опции с одного языка на другой. Например, *in Russian: table = таблица; move = переместить; edit = редактировать.*

Средства инспекции и отладки

Ясно, что отладка прикладной системы, объединяющей производственные правила, процедуры, различные уровни абстракции и иерархию классов, может превратиться в далеко не тривиальную задачу. В этой ситуации на помощь разработчику приходят мощные средства инспекции базы знаний и отладки. Функции инспекции базы знаний позволяют осуществлять поиск элементов на основе их типов, принадлежности к классу, атрибутов и местоположения. Эти функции используются для решения следующих задач:

- отображения сжатого представления элементов базы знаний;
- создания файлов, содержащих описание элементов базы знаний;
- отображения иерархий классов, модулей и рабочих пространств;
- прямого перехода к конкретным элементам базы знаний;
- перекомпиляции отдельных элементов.

В частности, упоминаемая выше схема иерархии встроенных классов G2 может быть получена с помощью функции инспекции базы знаний.

Перечисленные выше возможности облегчают навигацию по базе знаний и делают возможным ее быстрый просмотр под любым ракурсом. Кроме того, с помощью функций инспекции можно запустить процедуру поиска и замены текстовых фрагментов в базе знаний. Функции инспекции работают в фоновом режиме и позволяют выполнять одновременно с ними и другие задачи, включая другие функции инспекции. Доступ остальных пользователей к базе знаний в это время никак не ограничивается.

Отладочные режимы работы G2 включают:

- отображение предупреждающих сообщений об ошибках и состоянии приложения;
- отображение сообщений трассировки, которые показывают: текущие значения переменных и выражений, как только они изменяются; запуск и остановку обработки переменных, правил, формул и функций; выполнение каждого шага обработки переменных, правил, формул и функций;
- установку контрольных точек, в которых пользователь может остановить или продолжить процесс выполнения приложения;
- подсветку возбужденных правил.

Как показывает практика, функции инспекции и отладочные режимы работы G2 в значительной степени облегчают поиск ошибок в прикладной системе и сокращают время ее тестирования до минимума.

П1.4

Интерфейс с внешним окружением

В системе G2 реализована распределенная обработка приложения на принципах архитектуры “клиент-сервер”. Система Telewindows обеспечивает множественный доступ клиентов к централизованной базе знаний и групповую работу с приложением. Взаимодействие между G2 и Telewindows может быть организовано одним из следующих способов (рис. П1.4): процесс Telewindows выполняется на той же машине, что и G2, а пользователь получает к ней доступ через X-терминал; Telewindows выполняется на рабочей станции или персональном компьютере пользователя (Intel 386/486 - под MS Windows, v.3.1). Кроме того, приложение можно построить как содружество автономных интеллектуальных агентов на базе интерфейса $G2 \leftrightarrow G2$, исполняемых на одной и той же или на разных ЭВМ, связанных в сеть. При этом обмен данными осуществляется на уровне переменных через протокол ICP (Intelligent Communication Protocol). Интерфейс $G2 \leftrightarrow G2$ позволяет разработчику создать в одном G2-приложении объекты, которые получают информацию от другого G2. Эти объекты создаются подобно другим объектам в G2, а интерфейс $G2 \leftrightarrow G2$ действует, как сервер данных для этих объек-

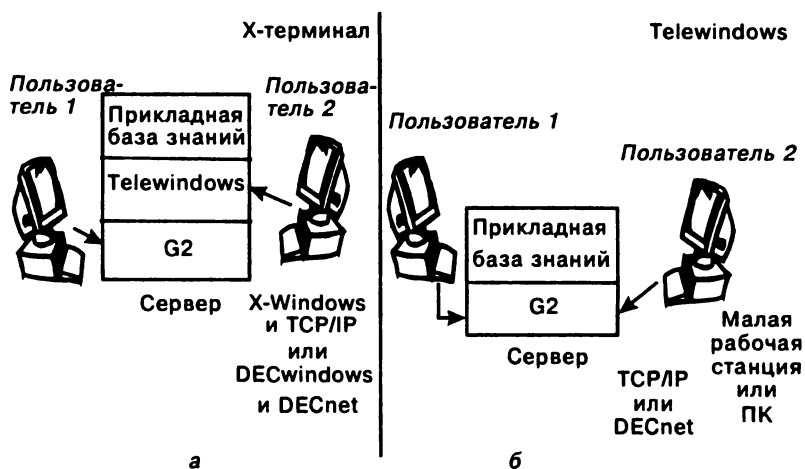


Рис. П1.4. Базовые конфигурации использования Telewindows:

а - процесс выполняется на том же ПК, что и система G2;

б - процесс выполняется на ПК пользователя

тов (подобно механизму вывода или G2-подсистеме моделирования). Для организации обмена необходимо в описании переменных объекта, получающих значение от другого G2-процесса, просто указать номер сетевого порта источника. Как результат, переменные объекта получают значения от второго G2. Подчеркнем, что G2-приложение может как получать, так и посылать информацию в одно и то же время по одному интерфейсу.

G2 разработана как открытая система. Связь с внешними источниками данных строится на основе библиотеки стандартных интерфейсов и сервера GSI (G2 Standart Interface). Подсистема GSI (рис.П1.5) работает параллельно с прикладной системой как независимый обработчик событий и обеспечивает ее двустороннее взаимодействие с широким спектром программируемых контроллеров ведущих фирм (Allen Bradley, GE-Fanuc, AEG Modicon), систем сбора данных (ABB, Fisher, Siemens, Yokogawa, Foxboro, ORSI), концентраторов данных (DEC BASEstar, Allen Bradley Pyrammid Integrator, SETPOINT SETCIM) и развитых СУБД (Oracle, Sybase, DEC Rdb). Библиотека GSI и так называемые G2 Bridge products позволяют легко интегрировать G2-приложение в существующие системы управления. При отсутствии в библиотеке GSI интерфейса к некоторому уникальному контроллеру не составляет особого труда запрограммировать его по представляемому шаблону на языке C и подключить к системе.

С точки зрения современной концепции разработки открытых систем в системе G2 предлагается более гибкая и надежная трех-звенная схема организации взаимодействия клиентов с сервером с ис-

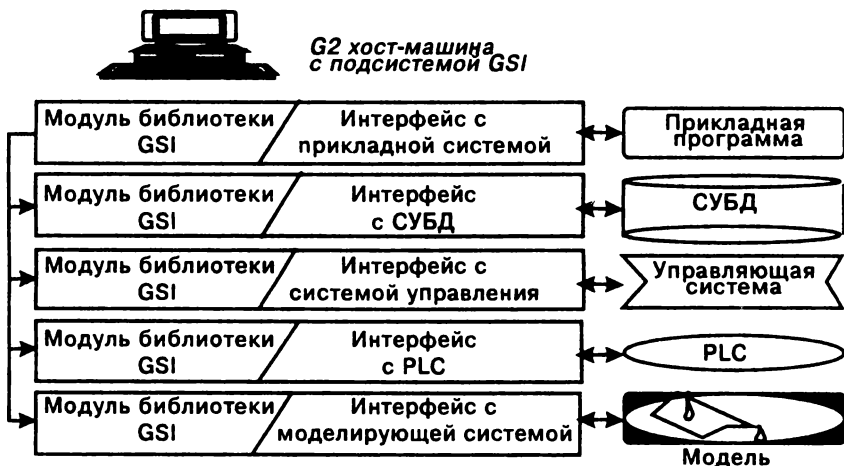


Рис. П1.5. Организация взаимодействия системы G2 с внешним миром

пользованием GSI в качестве монитора транзакций [1]. Обычно, когда перед разработчиком встает проблема создания интерфейса данных, он вынужден принимать во внимание целый ряд разнородных требований: эффективности, реализуемости, надежности, переносимости, сопровождаемости, конфигурируемости, гибкости, возможности мультиплексирования сигналов и т. д. Для удовлетворения этих требований он должен реализовать множество функциональных блоков, таких, как: синхронизация обрабатываемых запросов, протоколы взаимодействия; сетевые интерфейсы, восстановление после сбоев в сети или узле; работа с множеством источников данных; группировка данных; обработка данных, пришедших без запроса; обмен сообщениями об ошибках; обмен данными о состоянии взаимодействующих систем; буферизация данных; преобразование форматов данных, работа с данными переменной длины; планирование обработки запросов; отработка запуска и остановки системы; отработка пауз и перезапуска; отработка прерываний; обработка переполнения буфера; распределение ресурсов; минимизация загрузки системы; диагностика сбоев; доступ к внешним данным и конфигурирование интерфейса. Все это применимо к любому интерфейсу данных независимо от назначения прикладной системы. Все перечисленные требования, кроме двух, удовлетворяются в подсистеме GSI автоматически независимо от платформы и типа сетевого обеспечения. Исключение составляют функции доступа к данным и конфигурирование интерфейса, но реализацию этих функций подсистема GSI делает настолько простой, насколько это возможно.

Подсистема GSI состоит из трех основных частей: ядра GSI, GSI-расширения, коммуникационного канала связи между ядром GSI и GSI-расширением.

Взаимодействие этих трех частей между собой, с G2-приложением и внешней прикладной программой представлено на рис. П1.6.

Хотя подсистема GSI отрабатывает все взаимодействия между G2 и каждым из внешних процессов, необходимо сконфигурировать ее для конкретного приложения. Для разработки системы, в полном объеме использующей возможности GSI, нужно создать два фрагмента, отражающих специфику прикладной программы, в дополнение к базе знаний G2: спецификацию конфигурирования, которая настроит базу знаний для связи с внешней программой, и так называемый переходный код (application bridge code), который используется GSI-расширением для интерактивного взаимодействия с внешней прикладной программой.

Спецификация конфигурирования включает объекты базы знаний, конфигурирующие ее для использования GSI. Средства для создания этих объектов встроены в G2. Для формирования спецификации конфигурирования создаются объекты, принадлежащие к классу GSI interface object. Эти объекты содержат информацию, необходимую GSI для связи с внешней прикладной программой. Далее созда-

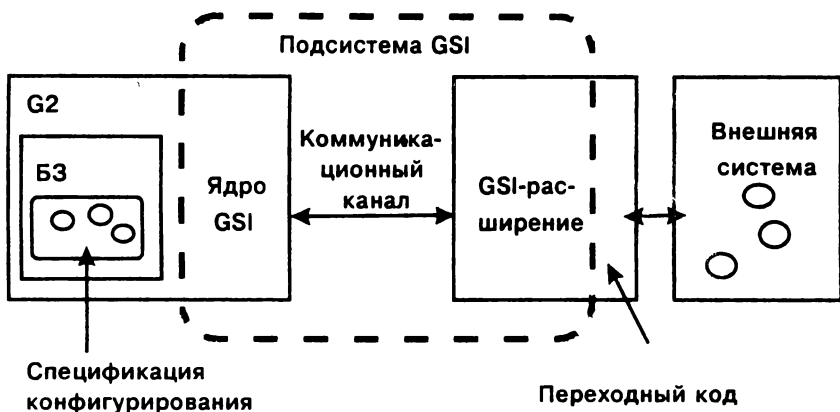


Рис.П1.6. Компоненты подсистемы GSI

ются переменные класса GSI variable, соответствующие переменным внешней прикладной программы. Система G2 передает и принимает данные через эти переменные. Кроме этого создаются переменные GSI variable для обмена текстовыми сообщениями с внешней прикладной программой.

Переходный код объединяет GSI-расширение с внешней прикладной программой. Он состоит из набора функций на языке C, обеспечивающих передачу данных, текстовых сообщений, запуск, остановку и завершение внешней программы.

Кроме интерфейсов GSI и $G2 \leftrightarrow G2$ доступны еще два интерфейса с внешними процессами и источниками данных: файловый интерфейс (G2 File Interface - GFI) и интерфейс с внешними функциями (Foreign Function Interface).

Файловый интерфейс GFI представляет собой гибкое средство, позволяющее G2 писать и читать информацию из внешних файлов. GFI является отдельным от системы G2 продуктом. Разработчик может использовать GFI для того, чтобы:

- архивировать и запоминать данные;
- инициализировать тесты проверки БЗ;
- собирать данные для внешнего анализа;
- создавать снимки (snapshots) данных;
- считывать данные из внешних файлов во время использования БЗ.

Интерфейс с внешними функциями включен в систему G2. Разработчик G2-приложения может вызвать внешние (foreign) для системы G2 функции, написанные на языках Си или Фортране. Для того чтобы использовать в G2 внешнюю функцию, разработчик описывает ее (подобно функциям G2, определенным пользователем) и затем использует таким же способом, как функции, определенные пользователем.

Возможности G2 4.0 в части поддержки распределенных приложений на основе архитектуры “клиент-сервер” расширены за счет добавления ряда новых функций. Передача объектов и массивов в качестве аргументов упрощает совместное использование данных независимыми приложениями на базе G2 и внешними по отношению к G2 программными системами. Для реализации обмена сложными структурами данных в распределенной среде обработки коренным образом переработан и G2 Standard Interface (GSI), служащий своего рода интеллектуальным монитором транзакций. Расширен также и спектр сетевых сред, в которых могут “жить” G2, Telewindiws и GSI. Кроме TCP/IP и DECnet теперь поддерживаются Token Ring и Winsock. Безопасность и конфиденциальность распределенной обработки достигаются за счет новых уровней автоматической проверки прав доступа при установлении сетевого взаимодействия процессов через GSI.

В версии 4.0 системы G2 в значительной мере сокращены накладные расходы на выполнение операций, что делает реальное время в приложениях на базе G2 еще более реальным. Подсекундные интервалы цикла работы Планировщика системы позволяют теперь контролировать процессы со скоростью протекания на уровне миллисекунд. Подсистема сбора профиля работы приложения дает возможность на этапе отладки и во время эксплуатации системы легко определять узкие места и целенаправленно проводить модификации для достижения требуемой производительности. Использование принципов статической и условной компиляции расширяет возможность маневра и дает возможность повышать производительность критических участков приложения при сохранении гибкости системы в целом.

Открытость системы G2 и продуктов, созданных на ее основе, обеспечивается ориентацией фирмы Gensym на промышленные стандарты. Являясь членом OMG (Object Management Group), фирма Gensym сотрудничает в этой области со многими независимыми организациями и комитетами по стандартам.

Ниже приведены стандарты, поддерживаемые системой G2.

Платформы: DEC, HP, SUN, IBM, ПЭВМ на базе X86 и Pentium, UNIX, VMS, Windows.

Графический интерфейс: Motif и (или) Windows, графический интерфейс GUIDE.

Сетевые протоколы: SNMP, TCP/IP, DECnet, WINSOCK.

Архитектура “клиент-сервер”: *уровень данных*: GSI, DDE, интерфейсы API; *уровень объектов*: CORBA; *уровень приложений*: Telewindiws.

Распределенная обработка: взаимодействие G2↔G2, G2↔Telewindiws.

Внешние интерфейсы: распространенные программные и технические средства и СУБД.

Участие в комитетах по стандартизации: **Object Management Group (OMG), POSIX, ISO 9000.**

В части технических средств - это поддержка широкого спектра платформ DEC, HP, Sun, IBM и ПЭВМ на базе процессоров X86 и Pentium. Развитый графический интерфейс, включающий элементы анимации, базируется на средствах Motif и MS Windows, сетевые протоколы - TCP/IP и DECnet. Архитектура "клиент-сервер" на уровне обмена данными поддерживается монитором транзакций GSI и DDE, на уровне объектов - CORBA, на уровне приложения - подсистемой Telewindows. Распределенная обработка обеспечивается интерфейсами $G2 \leftrightarrow G2$, $G2 \leftrightarrow Telewindows$ и поддержкой вызова удаленных процедур. Существует множество готовых решений "под ключ" для прямого взаимодействия G2 с распространенными программными и техническими системами контроля и развитыми СУБД.

П1.5

Проблемно(предметно)-ориентированные среды и графические языки на базе G2

Возможность простого манипулирования графическим представлением объектов в G2 и составления схем, являющихся отображением технологических цепочек или абстрактных алгоритмов обработки данных, обеспечивает базовые средства для построения проблемно-ориентированных языков визуального программирования. В этом случае объекты приобретают свойства операторов и в совокупности с различными классами связей формируют грамматику нового языка. Основным преимуществом такого подхода является то, что сформированная диаграмма потоков информации по сути и есть исполняемая программа, промежуточные фазы генерации кода и компиляции для ее использования не требуются. Впервые эта концепция была реализована фирмой Gensym в GDFL-языке графического представления информационных потоков для построения систем диагностики реального времени в системе GDA (G2 Diagnostic Assistant). Подход оказался настолько удачным, что в той или иной мере используется теперь во всех проблемно(предметно)-ориентированных расширениях G2.

Кроме базового продукта - оболочки ЭС реального времени G2 - фирмой Gensym разработаны дополнительные проблемно-ориентированные средства разработки. Основные из них: GDA, NeurOn-Line, ReThink (см. подразд.9.3), GST, Fault Expert и BatchDesign_Kit [3].

GDA (G2 Diagnostic Assistant) - это графический язык для вычислений и мониторинга, основанный на визуальных средствах описания решения задачи с пользовательским интерфейсом, организован-

ным по принципу WYSIWYG [5]. Объектно-ориентированный подход обеспечивает следующую технологию использования GDA: разработчик выбирает необходимые объекты из более чем сотни блоков, представляющих входные точки, фильтры данных, операции математической и логической обработки, управляющие воздействия и соединяет их.

Связи обеспечивают взаимодействие объектов. Программирование системы заключается в простом конфигурировании и связывании объектов.

Как и другие приложения на базе G2, система, созданная с помощью GDA, работает в реальном масштабе времени. GDA предназначена для решения следующих основных задач:

- системной диагностики, мониторинга и предупреждения о внештатных ситуациях (анализ и верификация сенсоров, фильтрация предупреждающих сигналов);
- управления качеством в реальном масштабе времени;
- предсказания и предупреждения внештатных ситуаций вместо их диагностики и борьбы с их последствиями (интеллектуальный интерфейс оператора, объяснения для определения исходной причины);
- упорядочивания управляющих воздействий и активного тестирования.

NeurOn-Line - проблемно-ориентированное расширение оболочки G2, предназначенное для решения задач многокритериального адаптивного управления на базе методологии нейронных сетей [3,7]. По принципам построения NeurOn-Line сходна с GDA. Приложение, использующее NeurOn-Line, представляет собой структуру, состоящую из узлов (нейронов) и связей между ними. Связи передают сигналы, могут усиливать или ослаблять их. Узлы осуществляют нелинейные преобразования сигналов. Вычисления на базе нейронной сети определяются: топологией сети, вычислениями в каждом из узлов, силой (весами) связей. Разработчику предоставляется широкий спектр наиболее распространенных типов нейронных сетей.

GST (Gensym Scheduling Toolkit) - инструментарий для построения систем оперативного планирования существует в двух вариантах. GST1 включает средства построения графиков Ганта, конфигурирования календаря и средства сортировки. GST2 является расширением GST1 и кроме перечисленных инструментов содержит Проектировщик расписаний, назначением которого являются:

- 1) обеспечение возможностей ввода: модели использования ресурсов; последовательности работ и начальных условий;
- 2) генерация расписания: определение недостаточности наличных ресурсов и несоответствия установленным срокам; получение оперативных корректировок и динамическое перепланирование расписаний.

Fault Expert - система, реализованная на базе G2. Является обобщением ряда прикладных систем в области управления телекоммуникациями:

- системы управления сетями ATM;
- диагностики спутников в INTELSAT;
- управления системой сообщений AT&T.

Fault Expert содержит репозиторий объектов, представляющих наиболее распространенное телекоммуникационное оборудование (различные типы компьютеров и терминальных устройств, модемы, маршрутизаторы, классы связей для представления физической среды передачи данных) и графический язык для описания процедур тестирования и сетевых протоколов прикладного уровня. Взаимодействие с распространенными сетевыми менеджерами HP OpenView, IBM NetView и DEC Policenter обеспечивается на уровне SNMP.

BatchDesign_Kit (BDK) предназначена для быстрого синтеза новых технологических процессов в фармацевтическом производстве. Использование BDK позволяет достичь баланса между стоимостью производства, объемом вредных отходов и производительностью.

В отличие от традиционных средств моделирования BDK нацелена на концептуальное проектирование технологического процесса во взаимодействии с фармакологами, химиками и технологами.

ЛИТЕРАТУРА

1. Кисель Е. Б. Сравнительный анализ инструментальных средств для разработки систем управления реального времени // Материалы семинара "Экспертные системы реального времени". - М.: ЦРДЗ, 1995.
2. Копченкова Н. В., Марон И. А. Вычислительная математика в примерах и задачах. - М.: Наука, 1972.
3. Попов Э. В., Фоминых И. Б., Кисель Е. Б., Шапот М. Д. Статические и динамические экспертные системы: Учеб. пособие. - М.: Финансы и статистика, 1996.
4. Попов Э. В. Экспертные системы реального времени // Открытые системы. - 1995. - №2.
5. A Guided Introduction to GDA. Gensym Corp., - Cambridge: (Mass., USA), 1992, November.
6. G2 Reference Manual. Version 4.0. Gensym Corp., - Cambridge: (Mass., USA), 1992, September.
7. NeurOn-Line Reference Manual. Gensym Corp., - NeurOn-Line Version 1.0 Beta Rev. 2. - Cambridge: (Mass., USA), 1993, July.
8. Smedley P. Who's Number One in Customer Service? // Control, 1994. November.

П2

ИМИТАЦИОННАЯ МОДЕЛЬ АВТОМАТИЗИРОВАННОГО ГРУЗОВОГО КОМПЛЕКСА



П2.1

Общие сведения

Имитационная модель технологического процесса обработки грузов (от приема груза с транспортного средства до выдачи его клиенту) разработана АО “Аргуссофт Компани” для проведения комплексных аналитических исследований эффективности функционирования подразделений и служб АО “Шереметьево-Карго”.

АО “Шереметьево-Карго” использует наличные ресурсы для выполнения трех основных технологических процессов.

Импорт - получение груза для клиента, извещение клиента, временное хранение груза до прихода клиента, выдача груза клиенту;

Экспорт - прием груза от клиента, временное хранение груза, отправка груза в пункт назначения;

Трансфер - получение груза для дальнейшей транспортировки, временное хранение груза, отправка груза в пункт назначения.

Модель отображает использование ресурсов предприятия при совместном исполнении перечисленных процессов. Используемые при разработке средства моделирования обеспечивают:

- отображение и сбор статистики по выбранной системе технико-экономических показателей;
- введение новых показателей;
- модификацию модели для отображения новых схем исполнения технологического процесса;
- интерфейс с пользователем для изменения значений регулируемых параметров.

В состав регулируемых (изменяемых) параметров модели включены:

- характеристики грузопотока;
- количество и структура людских ресурсов;
- количество и номенклатура используемого оборудования;
- структура затрат на исполнение моделируемого процесса.
- Изменяемые характеристики модели, позволяющие оценивать результаты деятельности предприятия, включают:

- сводные показатели эффективности исполнения технологического процесса;
- себестоимость обработки грузов в ходе исполнения моделируемого процесса;
- коэффициенты использования оборудования и людских ресурсов;
- время обработки грузов.

Модель обеспечивает возможность, варьируя значения регулируемых параметров, получать сводные характеристики исполнения моделируемых процессов.

Экспериментальная апробация модели позволила уточнить структуру и состав необходимых параметров для разработки полной модели автоматизированного грузового комплекса (АГК), включающей:

- комплекс технико-экономических показателей, позволяющих объективно оценивать использование ресурсов и оборудования, а также удовлетворять требования клиентов АГК в условиях моделируемой организационной структуры;
- набор регулируемых параметров модели, отображающих как внешние условия функционирования предприятия (расписание прибытия и отправки авиарейсов, налоговые ставки и т.п.), так и стратегические управляющие воздействия (тарифные ставки, запасы ресурсов).

Методика проведения экспериментов

На верхнем уровне моделирования (уровень детализации 0) АГК представляется в виде единого блока с двумя входами и тремя выходами (рис. П2.1). Ряд экономических показателей блока АГК характеризуют затратные статьи, не зависящие от технологических характеристик модели. Показатель ВЕС-ОБРАБОТАННОГО-ГРУЗА накапливает нарастающим итогом вес грузов, покинувших АГК с начала моделирования. Показатель ТАРИФ позволяет установить расценки АГК на килограмм груза, оплачиваемого клиентами, для определения дохода предприятия. Подтаблица СТОИМОСТЬ содержит стоимостные статистики, рассчитываемые в ходе моделирования. Наибольший интерес для исследования представляет характеристика ОБЩАЯ-СТОИМОСТЬ, содержащая текущий уровень затрат на предприятии, исчисляемый из затрат на использование ресурсов при исполнении отдельных технологических операций.

Входы модели

Прибывающие авиарейсы - данный вход разделяется на два потока:

- авиарейсы, поступающие на перроны АГК и Шереметьево-2 (Ш-2);
- информации о рейсах, поступающих на АГК за два часа до прибытия соответствующего рейса.

Клиенты - отправители грузов, являющиеся источником информации об объеме отправляемого груза.

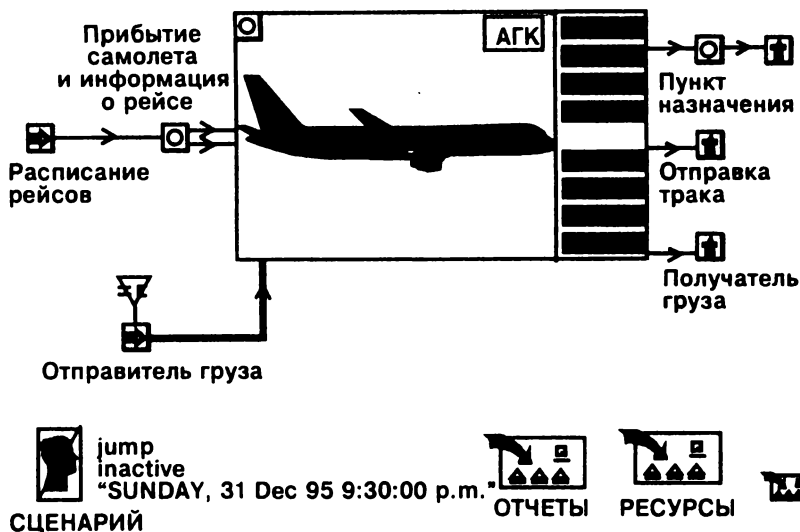


Рис.П2.1. Модель технологического процесса обработки грузов

Выходы модели

Пункт назначения. Самолет, покидая блок АГК, попадает в пункт назначения, где из него извлекается экспортный груз. Самолеты и грузы покидают модель. Дополнительная функция, исполняемая данным блоком, - генерация грузов, отправляемых на АГК зарубежными клиентами.

Отправка траков - импортные грузы, предназначенные для дальнейшей транспортировки внутренними авиарейсами, покидают АГК на автотраках.

Клиент-грузополучатель - через этот блок грузы, получаемые клиентами в Москве, покидают модель.

П2.2

Использованные инструментальные средства

Имитационная модель автоматизированного грузового комплекса разработана на основе инструментального комплекса ReThink (см. подразд. 9.3) фирмы Gensym (США), позволяющего лицу, принимающему решения при управлении большой компанией, визуализировать процесс делопроизводства и анализировать его производительность.

После выяснения на модели текущего положения дел в компании аналитик может ставить эксперименты с альтернативными схемами организации работ для решения следующих задач:

- снижение затрат;
- сокращение времени обслуживания клиентов;
- повышение качества исполнения работ;
- стандартизация исполняемых функций.

ReThink обеспечивает широкий спектр функциональных возможностей, ориентированных на моделирование организационных и технологических процессов.

Графическая среда разработки и механизмы моделирования

ReThink поддерживает интерактивную разработку моделей организационных и технологических процессов. Он обеспечивает графическую среду проектирования бизнес-моделей, объектно-ориентированную подсистему имитации для тестирования этих моделей и инструментарий для установления и измерения временных, стоимостных и других показателей эффективности деятельности компании.

Моделирование бизнес-процессов

С помощью системы ReThink можно моделировать любые типы деловой активности. Например, ReThink успешно применяется для моделирования и анализа следующих типов бизнес-процессов:

- процесс оформления заказов;
- производство и доставка;
- продажный цикл и производительность;
- цикл исследований и разработки новой продукции;
- снабжение и оплата счетов и т.д.

Анимация

ReThink автоматически “оживляет” разработанную модель, облегчая визуализацию исполнения любой сложной деятельности. Если необходимо получить только обобщенные характеристики моделируемого процесса, пользователь имеет возможность отключения средств анимации для увеличения производительности.

Автоматический анализ производительности

ReThink обеспечивает автоматическую фиксацию метрик анализируемого процесса. Эта возможность позволяет проводить эксперименты с различными организационными и производственными структурами, сравнивая их по стоимостным и временным характеристикам. Кроме того, ReThink предоставляет широкий спектр средств для проведения факторного анализа. Например, достаточно быстро можно определить, как добавление ресурсов в ключевых точках процесса повлияет на его производительность.

Основные компоненты ReThink

ReThink включает ряд базовых компонентов, на основе которых строится модель бизнес-процессов (см. разд. 9.3).

• **Сценарии** управляют механизмами моделирования дискретных событий и позволяют проводить одновременное исполнение нескольких моделей. В описываемой модели использован единствен-

ный сценарий, с помощью которого пользователь может управлять процессом моделирования АГК.

- **Блоки** выполняют такие операции над объектами, как создание объектов, исполнение бизнес-функций, установление и разрыв ассоциаций между объектами, удаление объектов.

- **Инструменты** позволяют получать и отображать в числовой и графической форме данные о производительности моделируемых процессов. Кроме того, инструменты позволяют в заданных точках модели вводить необходимые значения параметров.

- **Ресурсы** (средства труда) предназначены для ограничения исполняемых операций на основе объема и состава наличных ресурсов.

- **Рабочие объекты** (предметы труда) проходят через блоки модели и обрабатываются ими, аккумулируя статистики производительности в каждой точке моделируемого процесса.

П2.3

Структура модели

Основные элементы модели приведены на рис. П2.2.



Рис. П2.2. Основные элементы модели

П2.3.1

Рабочие объекты (предметы труда)

Рабочие объекты, используемые в модели, являются производными от класса бпр-объект и наследуют от него стандартные стоимостные и временные характеристики. Общим для представителей класса бпр-объект является накопление в атрибуте ОБЩАЯ-СТОИМОСТЬ подтаблицы СТОИМОСТЬ затрат на исполнение всех операций, в которых участвовал объект, т.е. себестоимости его обработки. Ниже приводятся основные рабочие объекты модели АГК с краткой характеристикой их назначения и специфических атрибутов.

Самолет характеризуется номером авиарейса, временем прибытия и вылета, средним и максимальным количеством доставляемого груза. Характеристики авиарейсов, используемые в модели, задаются в файлах РАСПИСАНИЕ и ИНТЕРВАЛЫ. В первом из них содержится информация о рейсе, а во втором - интервалы времени между прибытием рейсов в Москву. Для учета того, что информация о прибытии рейса поступает на АГК за два часа до реального прибытия и для учета разницы между прибытием последнего и первого рейсов в расписании установлен сдвиг первого рейса на 1 час. С другой стороны, для учета недельного цикла расписания стартовое время модели должно устанавливаться в 21ч30мин любого воскресенья. При соблюдении указанных условий рейсы в модели будут прибывать в соответствии с реальными временами и днями недели. На каждый рейс отводится одна строка файла, в которой перечислены имена атрибутов и желаемые значения. При необходимости файлы могут быть отредактированы любым текстовым редактором. Единственное ограничение - символы кириллицы должны вводиться в соответствии с кодовой таблицей ISO Latin 1.

Информация о рейсе содержит информацию о реальном объеме груза, прибывающего с данным авиарейсом на АГК, и перроне, на который прибывает рейс. Данная информация служит основой для формирования сцепки тележек, подаваемых на соответствующий перрон под разгрузку.

Команда на вылет формируется в момент прибытия рейса на АГК и запускает таймер, гарантирующий отбытие рейса из Москвы в соответствии с расписанием. Таймер работает параллельно с операциями разгрузки и погрузки самолета и поэтому позволяет зафиксировать отклонения от графика работ, ведущие к задержке вылета рейса.

Авианакладная в модели является основным документом, сопровождающим груз во время обработки. Общий вес груза на накладную определяется по статистикам, характерным для реального грузопотока на АГК.

Груз является элементом обработки и хранения в ячейках складов АГК. Характеризуется весом и ассоциирован с накладной, по которой он прибыл.

Клиент - в явном виде в модели представлен только клиент - отправитель груза. После прибытия клиента в блок бронирования в модели появляется авианакладная, клиент покидает модель, и дальнейшие операции осуществляются с накладной.

Тележка вместе с палетой используется для перевозки грузов к самолету и от самолета на склад.

Палета в модели используется для перевозки грузов к самолету и от самолета на склад. Вместе с грузом и комплектом авианакладных отправляется в пункт назначения и прибывает на АГК с самолетом.

Комплект - абстрактный объект, служащий для объединения авианакладных, лежащих на одной палете во время транспортировки.

Сцепка - абстрактный объект, служащий для объединения от одной до шести тележек, перевозимых трактором по территории АГК и между АГК и перроном Ш-2.

П2.3.2

Ресурсы (средства труда)

Ресурсы являются производными от класса бпр-ресурсы и предназначены для ограничения исполняемых операций на основе объема и состава наличных ресурсов. Каждый из ресурсов характеризуется стоимостью его использования, переносимой в процессе моделирования в затраты на исполнение соответствующих операций, и себестоимостью обработки рабочих объектов.

Трактор служит для транспортировки сцепки тележек по территории АГК и между АГК и перроном Ш-2.

Автотрак служит для транспортировки грузов с авианакладными, отправляемых из других аэропортов Москвы.

Грузчик привлекается для выполнения операций, связанных с погрузкой (разгрузкой) транспортных средств и помещением (выдачей) грузов на хранение.

Ячейка - единица хранения в складских помещениях.

На рис. П2.3 приведена детализация технологического процесса обработки грузов на АГК.

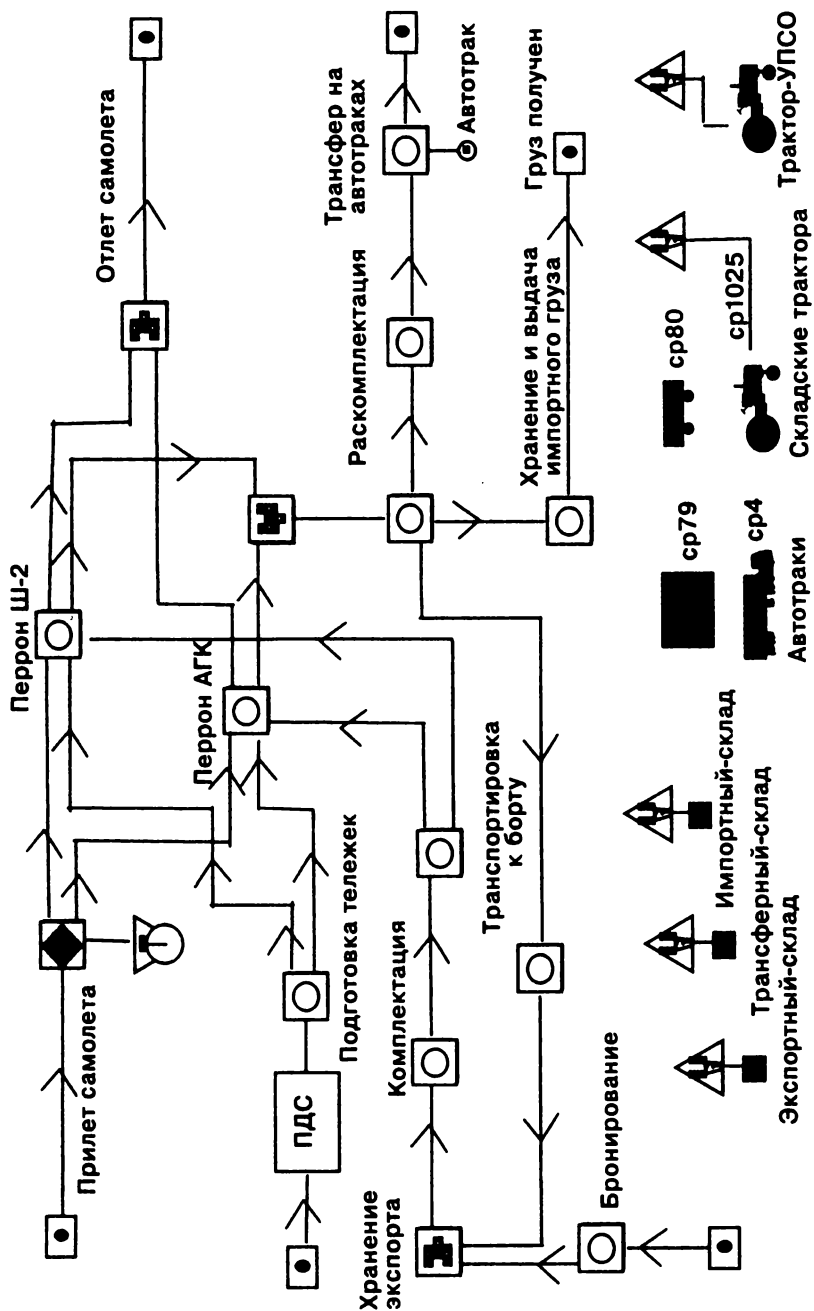


Рис.П2.3. Детализация технологического процесса обработки грузов на АГК

П2.4

Работа с моделью

В данном разделе рассмотрен типичный порядок запуска и экспериментирования с моделью.

П2.4.1

Загрузка базы знаний модели

Для загрузки базы знаний модели в главном меню системы G2 выбирают пункт *Загрузка*. В появившемся диалоговом окне вводят путь к каталогу, содержащему файлы базы знаний модели. Главным модулем, который следует выбрать для загрузки, является *Модель* (стартовое состояние модели) или *М.снимок* (моментальный снимок работающей модели, сохраненный на предыдущем этапе экспериментов).

После загрузки базы знаний модели автоматически стартуют модули ReThink.

П2.4.2

Отображение головного рабочего пространства модели

Из главного меню системы G2 необходимо выбрать пункт *Get Workspace*. Из списка поименованных рабочих пространств нужно выбрать *TUTORIALS*. Головное рабочее пространство модели является подпространством блока *Модель*.

П2.4.3

Активация сценария

В меню сценария модели (блок с рисунком головы в профиль в нижнем левом углу рабочего пространства) необходимо выбрать пункт *activate-scenario*. На экране автоматически появится подпространство сценария (рис. П2.4), позволяющее инициализировать модель и установить начальные значения параметров моделирования.

Status	Mode	User	
Continue	Jump	Admin	Start Time ****
Pause	Step	Developer	Current Time ****
Reset	Synch	User	Elapsed Days ****

Activation	Animation	
Activate	Enable	Blocks ****
Deactivate	Disable	Objects ****
		Scheduled Events ****

Моментальный снимок

График интенсивности прилета самолетов

20

▼ 3600.0

3600.0

5

▼ 3600

600.0

Количество автотраков	Количество складских помещений	Количество тележек на перроне АВК Ш-2	Количество тележек на УПС
4	9	10	60

Рис. П2.4. Панель управления сценарием

П2.4.4

Установка начальных значений параметров моделирования

На данном уровне детализации АГК характеризуется объемами грузов, поступающими на входы и покидающими модель через перечисленные выходы. Экспериментатор имеет возможность устанавливать напряженность грузопотока, изменяя расписание рейсов и интервалы времени между поступлением экспортных и импортных грузов. В модели предусмотрены два способа задания расписания авиарейсов:

- на основе данных, записанных в текстовых файлах РАСПИСАНИЕ (характеристики 230 реальных рейсов, прибывающих на перроны Шереметьево-2 и АГК в течение недели) и ИНТЕРВАЛ (временные интервалы между прибытием соответствующих рейсов);
- на основе графика интенсивности прилетов, находящегося в подпространстве сценария модели.

Как в первом, так и во втором случае характеристики рейсов (средний вес привозимых грузов, место разгрузки, длительность стоянки перед обратным вылетом) берутся из файла РАСПИСАНИЕ.

В подпространстве сценария находятся также ползковые регуляторы, позволяющие установить интервал между поступлением импортных и экспортных клиентов. Предполагается, что клиенту соответствует одна авианакладная, вес грузов и количество мест на накладную определяются в соответствии со статистиками реального грузопотока.

Установив исходные характеристики грузопотока, не покидая подпространства сценария, экспериментатор имеет возможность изменить количество ресурсов, привлекаемых для выполнения процессов обработки грузов: количество автотраков; количество палет на складе УПСО; количество складских тракторов; количество тележек на УПМ и перроне Ш-2; количество складских тракторов.

После ввода числовых значений в соответствующие поля для изменения параметров модели необходимо нажать кнопку Установить повторно. После инициализации блоков модели указанная кнопка вызывает процедуру НАЧАЛЬНЫЕ-ЗНАЧЕНИЯ, устанавливающую соответствующие параметры моделирования и инициализирующую некоторые служебные переменные.

Запуск модели

Следующим этапом после установки начальных значений является запуск модели. Запуск может быть осуществлен в одном из трех режимов: дискретный, пошаговый и синхронный.

В режиме *дискретный (Jump)* на каждом шаге моделирования ReThink формирует список ближайших событий модели (прибытие самолета, помещение груза в ячейку склада и т.п.), выбирает из него ближайшее по времени событие, осуществляет перевод часов модели на момент наступления данного события и пересчитывает стоимостные и временные характеристики блоков, рабочих объектов и ресурсов. После этого процесс повторяется. Таким образом, переход между двумя последовательными событиями модели осуществляется мгновенно.

Пошаговый (Step) режим работы модели отличается от режима с переменным шагом только тем, что после отработки каждого события модель останавливается и для продолжения процесса пользователь должен нажать кнопку *Continue* в подпространстве сценария или выбрать пункт *resume-scenario* в меню сценария. Пошаговый режим используется в основном для отладки модели или для детального изучения какого-либо процесса.

В *синхронном (Synch)* режиме работы ReThink масштабирует временные интервалы между соседними событиями модели в соответствии с заданным коэффициентом сжатия (час в секунду по умолчанию). Это позволяет добиться большей наглядности в процессе моделирования и равномерности временной шкалы. Однако отрабатываются все события модели, даже если интервал между ними меньше заданного временного шага. Поэтому синхронный режим работы является более медленным по сравнению с режимом с переменным шагом. Желаемый режим работы выбирается с помощью соответствующей кнопки в подпространстве сценария модели.

Непосредственно запуск модели осуществляется с помощью кнопки *Continue* в подпространстве сценария или выбора пункта *resume-scenario* в меню сценария. Для того чтобы рабочие объекты начали поступать в модель из блоков-источников, необходимо выбрать пункт меню сценария *start-всех*. (Для анализа влияния отдельных источников или любой их комбинации можно выбрать пункт меню *start* соответствующего блока-источника.)

Наблюдение за ходом моделирования

Система ReThink и лежащий в ее основе инструментальный комплекс G2 обеспечивают широкий спектр форм представления информации для наблюдения за ходом моделирования.

Причинно-следственные взаимосвязи, взаимовлияние параллельных потоков событий и порядковые отношения между наступлением событий во времени экспериментатор имеет возможность отслеживать непосредственно на диаграммах модели, наблюдая за перемещением рабочих объектов модели между блоками и за занятием и освобождением соответствующих ресурсных объектов.

Динамика изменения различных показателей отображается на графиках, помещенных в рабочем пространстве ОТЧЕТЫ (рис. П2.5). Здесь присутствуют графики удельной загрузки складов, коэффициентов использования тракторов, выходного грузопотока, коэффициента использования тележек и графики экономических показателей. Графики удельной загрузки складов и коэффициентов использования тракторов реализованы полностью на базе средств системы ReThink. Графики выходного грузопотока реализованы в рамках концепций ReThink, но базируются на дополнительном универсальном блоке извлечения данных. Графики коэффициентов использования тележек и экономических показателей базируются на возможностях, предоставляемых G2. Частично это сделано для демонстрации возможностей гибкого объединения приложений на базе G2 с моделями ReThink.

Текущее состояние ряда показателей приведено в рабочем пространстве РЕСУРСЫ (табл. П2.1). В этом пространстве помещены две таблицы: занятости наличных ресурсов и очередей на выполнение ключевых операций. Таблицы реализованы на основе базовых средств G2 - таблиц свободной формы.

Кроме таблиц, графиков и диаграмм модели можно использовать свойства всех элементов модели как объектов G2 (см. П1). В этом качестве все рабочие объекты, блоки и ресурсы обладают собственными таблицами атрибутов, отражающими их текущее состояние. К этим таблицам можно получить доступ через пункт меню **таблица** соответствующего объекта. При необходимости можно непосредственно изменять атрибуты объектов, определяющие характеристики модели, через эти таблицы.

РЕСУРСЫ

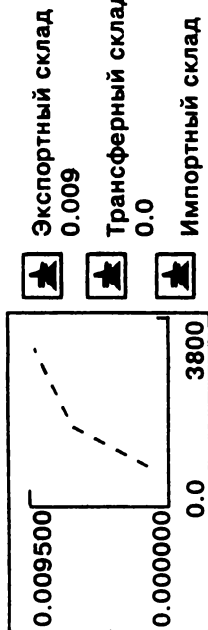
Время	"SUNDAY, 31 Dec 95 9:37:15 p.m."
-------	----------------------------------

Участок	Ресурс	Всего	Занято
Перрон АВК Ш-2			
	ГРУЗЧИКИ	30	8
	ТРАКТОРА	11	1
	ТЕЛЕЖКИ	10	---
Склад импортный			
	ГРУЗЧИКИ	45	31
	ТРАКТОРА	9	2
	ЯЧЕЙКИ	19204	251
Склад экспортный			
	ГРУЗЧИКИ	10	5
	ЯЧЕЙКИ	933	135
Склад трансферный			
	ГРУЗЧИКИ	4	1
	ЯЧЕЙКИ	960	180
УПМ			
	ТЕЛЕЖКИ	70	12
УПСО			
	ТРАКТОРА	3	1
	ПАЛЕТЫ	8	2

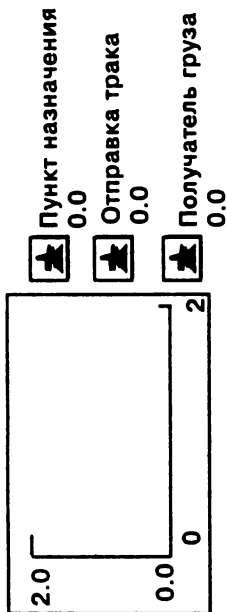
ОЧЕРЕДИ

Разгрузка на Ш-2	0
Вылет на Ш-2	0
Разгрузка на АГК	0
Вылет на АГК	0
Раскомплектация	0
Накладные, груз по которым не получен	0

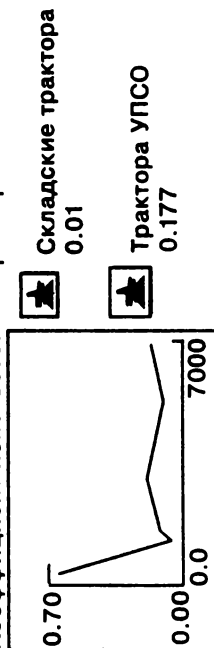
Удельная загрузка складов



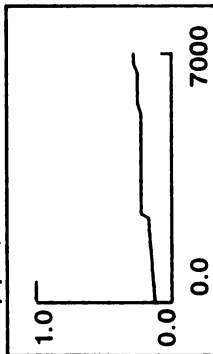
Выходной грузопоток



Коэффициент использования тракторов



Коэффициент использования тележек



Относительная технологическая себестоимость

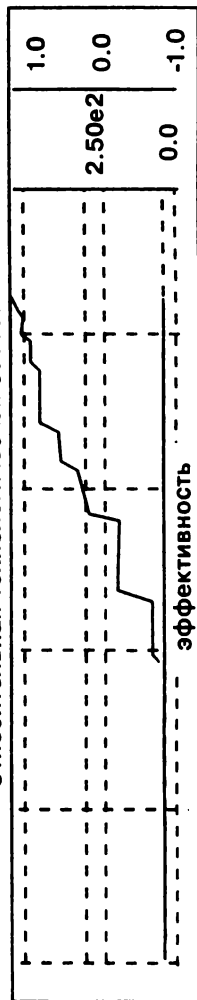


Рис. П2.5. Графики, генерируемые моделью АГК

П2.4.7

Моментальный снимок текущего состояния модели

Кнопка *Моментальный снимок* служит для архивирования текущего состояния модели с возможностью в дальнейшем “теплого пуска” процесса моделирования с данной контрольной точки. Моментальный снимок состояния сохраняется в файле *Моментальный снимок* в текущем каталоге модели. Переписывая данный файл или переименовывая его, можно создать набор контрольных точек, наиболее интересных с точки зрения анализа протекания моделируемых бизнес-процессов.

П2.5

Перспективы дальнейшего развития модели

В дальнейшем предполагаются развитие и детализация модели в направлении повышения ее адекватности реальному процессу обработки грузов АО “Шереметьево-Карго”, что позволит оценивать результаты перераспределения ресурсов и структурной реорганизации работы предприятия до их внедрения, а также эффективность работы с клиентами (в частности, тарифной политики) в условиях современного динамично развивающегося рынка транспортных и складских услуг. В более отдаленной перспективе планируется преобразование модели в систему поддержки принятия решений диспетчерским персоналом АГК, интегрированную в АСУ “Шереметьево-Карго” и призванную оптимально распределять поступающие самолеты и грузы, а также оптимально выделять наличные ресурсы рабочей силы и техники в реальном масштабе времени.

ПРЕДМЕТНЫЙ УКАЗАТЕЛЬ

А

Автор 120
Агрегат объектов 99, 139
Арбитр 120
Анализ требований 204, 212, 213
Атрибут 142

Б

Бета-тестирование 21, 223, 224
Бизнес-процесс 16
Бизнес-система 159
Блоки 267
 Associate (ассоциация) 267
 Batch (упаковка) 267
 Branch (ветвление) 267
 Copy (копирование) 267
 Copy Attributes (копирование атрибутов) 268
 Merge (слияние) 267
 Reconcile (ожидание пары) 267
 Sink (приемник) 267
 Source (источник) 266
 Task (задача) 267
 Insert (вставка) 268
 Remove (удаление) 268
 Store (сохранение) 268
 Retrieve (извлечение) 268

В

Вертикальное сжатие процессов 50
Виртуальная команда 55
Владелец процесса 68
Владелец ресурса 68
Возвратный метод проектирования 89

Г

Горизонтальное сжатие процесса 50
Группа 66
 документирования 67
 обеспечения качества 66
 прототипирования 67

Д – 3

Дедуктивное мышление 44
Диаграммы Мили 142
Зонд (probb) 270

И

Идеальное проектирование 203, 204
Иерархия классов 84, 277
Индуктивное мышление 44
Инженерия знаний 95, 242
Инжиниринг бизнеса 12, 16, 18
Инкапсуляция 85, 86
Инспектор 120
Инспекция 119
Инструмент 258, 270
Интерфейсный объект 137, 189
Информационная система 43
Информационные технологии 12, 25
Источник данных (сервер данных) 283
Итератор 285

К

Каскадный метод 83
Класс 85, 86
 описаний (definition) 281
 описание объекта (object-definition) 281
 описание связи (connection-definition) 281
 описание сообщения (message-definition) 281
Классы 281
 описаний классов (class definition) 281
 пользовательского интерфейса (user-interface) 281
 языка G2 (G2 language) 281
Клиент 12, 129, 167
Команда по реинжинирингу 65, 66
Координатор (или группа координации) 67

Л

Лидер 64
 проекта 64, 65
 процесса 69

М

Метрики 103, 171

Модель 72, 79

 бизнеса 74, 77, 80

 внешняя модель 125, 127

 внутренняя модель 125

 идеальная 135, 188

 идеальная О-модель 218, 177, 187

 О-модель (объект-модель) 135, 143, 172

 П-модель (прецедент-модель) 127

 реальная модель 135, 188

 реальная О-модель 177

Моделирование 81

 имитационное 81, 95, 244

 объектно-ориентированное 95

Модуль 278

Наследование 85, 86, 89

Непосредственно требуемый модуль 278

О

Обсуждение 119, 121

 неформальное 119

 формальное 118

Объект 280, 282

 временный 282

 идеальный 210

 рабочий 266

Объект (object) и его подклассы 280

Объектно-ориентированный подход 84, 85, 89, 244

Объекты-сущности 137

Одновременная инженерия 24

Оператор процесса 60

Операторы G2 297, 305

Операция 281

 изменить текст (change the text of) 281

 сделать объект постоянным (make permanent) 281

 создать объект клонированием (create by cloning) 281

Описание 73

 динамическое 73

 статическое 73

Организационное прототипирование 198

О

- Ориентация на процесс 24
- Отношение 284
 - "зависит от" 152
 - использования 148
 - коммуникации 140, 228
 - наследования 192
 - расширения 148
 - ссылки 139, 192

П

- Параметр 283
 - операции 141
- Передачик (remote) 271
- Переменная 283
- Пиктограмма (икона) 282, 293
- Подбизнес (sub-business) 124, 152
- Подсистема 152, 160
- Полиморфизм 85, 87
- Поток событий 73, 132, 185, 219
- Поток ценностей 18
- Правило 23
 - общее 284
 - специфическое 284
- Прецедент 127, 129
- Прогон (Walkthrough) 120
- Продукция 20, 125
- Протокол класса 141
- Процесс 11, 13

Р

- Рабочее пространство БЗ (Kb-workspace) 280
- Расширяемая конструкция 148, 221
- Реализация 203
- Реальное проектирование 203, 218, 236
- Реинжиниринг бизнес-процессов 12, 13, 42
- Ресурсы 258, 271
- Руководящий комитет 64, 65

С

- Сбор требований 203
- Связь (connection) 280, 283
- Секретарь 120
- Сигнатура операции 141
- Спиральный подход 83

С

Средства быстрой разработки приложений 245
Стандартные атрибуты рабочих объектов 269
Стратегическая цель ИТ 49
Субъект 127
Сценарий 81, 258, 311

Т – У

Тестирование 203
Управляющий объект 208
Уровень 158
 бизнес-системы 158
 инфраструктуры 158, 160
 общих бизнес-объектов 158
 ресурсов 158
Установка 299
Установщик (feed) 258, 270

Ф – Ш

Функция 124
Штат менеджера проекта 67

Э

Эксперт по методу 66
Этапы эволюции БПР 43, 44
 начальный (тактический) период БПР 43
 глобальный период БПР 44
 стратегический период БПР 44

BatchDesign_Kit (BDK) 307
CASE-технологии 95, 249
G2 274
GDA (G2 Diagnostic Assistant) 265, 275, 305
Gensym 245
GST (Gensym Scheduling Toolkit) 306
NeurOn-Line 306
ReThink 127, 311, 258, 269

ОГЛАВЛЕНИЕ

К читателю 3

Предисловие 7

1

Введение в проблему 11

1.1

**Реинжиниринг бизнеса - новое направление
теории менеджмента 11**

1.2

Примеры успешного реинжиниринга 20

1.3

Условия успешного реинжиниринга и факторы риска 25

1.4

Что не является реинжинирингом 35

1.5

Ответы на типичные вопросы о реинжиниринге 36

2

**Реинжиниринг
и его воздействие на компанию 41**

2.1

**Инжиниринг бизнеса, реинжиниринг бизнеса
и усовершенствование бизнеса 41**

2.2

Роль информационных технологий в реинжиниринге 44

2.3

Особенности перепроектированных бизнес-процессов 49

2.4

Последствия реинжиниринга бизнес-процессов 54

2.5

Структура традиционной и новой компании 60

2.6

Участники проекта по реинжинирингу и их роли 64

3

Моделирование бизнеса 72

3.1

Модель бизнеса 72

3.2

Требования к модели компании 77

3.3

Создание модели бизнеса 81

3.4

Традиционные способы разработки моделей 82

3.5

Объектно-ориентированный подход к разработке моделей 84

3.6

Интегрированные подходы к разработке моделей 94

4

Основные этапы реинжиниринга бизнес-процессов 98

4.1

Структура и этапы реинжиниринга 98

4.2

Основания для начала работ по реинжинирингу 104

4.3

Разработка образа будущей компании 105

4.4

Спецификация целей компании 109

4.5

Разработка модели существующего бизнеса 110

4.6

Разработка модели нового бизнеса 111

4.7

Внедрение новых процессов 112

4.8

Управление проектом по реинжинирингу бизнеса 112

5

Архитектура инжиниринга бизнеса 123

5.1

Требования к бизнес-модели 123

5.2

Внешняя модель 127

5.3

Внутренняя объектная модель 134

5.4

Прецеденты и объекты 143

5.5

Отношения между прецедентами 148

5.6

Подсистемы и прецеденты 152

5.7

Масштабирование моделей бизнеса 153

6

Модель существующего бизнеса 162

6.1

Назначение обратного инжиниринга 162

6.2

Общая картина бизнеса 162

6.3

П-модель существующего бизнеса 166

6.4

О-модель существующего бизнеса 172

6.5

Анализ результатов 174

**Модель
нового бизнеса 177**

7.1

П-модели нового бизнеса 177

7.2

О-модели нового бизнеса 177

7.3

Диаграммы взаимодействия 194

7.4

Разработка информационной системы 197

7.5

Проверка нового бизнеса 197

8

**Построение
информационной системы поддержки 200**

8.1

Модели бизнес-процессов и информационные системы 200

8.2

Разработка программного обеспечения 201

8.3

Объекты бизнес-системы,
разрабатывающей программное обеспечение 210

8.4

Разработка информационной системы
и разработка бизнеса 224

8.5

Внедрение новой информационной системы 233

**Методологии и инструментальные средства
для проведения реинжиниринга 242**

9.1

Методологии реинжиниринга 242

9.2

Классификация и анализ ИС 245

9.3

ReThink - интегрированный многофункциональный комплекс
для проведения реинжиниринга 257

Приложения

П1

G2 - Инструментальный комплекс
для создания динамических интеллектуальных систем
в управлении и моделировании 274

П2

Имитационная модель
автоматизированного грузового комплекса 308

Предметный указатель 324

Об авторах

ОЙХМАН Ефим Григорьевич

родился в 1938 г. в Пензе. В 1962 г. закончил Пензенский политехнический институт по специальности “Математические и счетно-решающие приборы и устройства”. В конце пятидесятых годов начал работать в знаменитом филиале СКБ 245 под руководством Б.И. Рамеева, где были созданы первые отечественные ЭВМ “Урал”. После окончания института более 15 лет занимался на предприятиях Министерства радиопромышленности СССР разработкой и применением специализированных ЭВМ. Работая в ИНЭУМ под руководством Б.И. Наумова, защитил докторскую диссертацию, получил звание профессора.

В 1991 г. организовал Российский НИИ информационных технологий и систем автоматизированного проектирования, где работает по настоящее время директором. С 1968 г. преподает в Московском институте радиоэлектроники и автоматики (МИРЭА). Заведует базовой кафедрой. Автор более 118 печатных работ и 5 монографий.

ПОПОВ Эдуард Викторович

родился в 1940 г. в Ленинграде. В 1963 г. окончил Ленинградский институт точной механики и оптики по специальности “Математические и счетно-решающие приборы и устройства”. Работал в Ленинградском КБ под руководством Ф.Г. Староса – основателя отечественной микроэлектроники, где занимался разработкой трансляторов для языков программирования и языков искусственного интеллекта.

С 1974 по 1991 г. работал на предприятиях Министерства радиопромышленности СССР, где занимался системами общения с ЭВМ на русском языке и экспертными системами. В 1982 г. защитил докторскую диссертацию, а в 1995 г. получил звание профессора.

С 1991 г. по настоящее время работает в Российском НИИ информационных технологий и систем автоматизированного проектирования, где разрабатывает экспертные системы, системы моделирования производственных и организационных процессов (бизнес-процесс реинжиниринг).

Является вице-президентом Российской Ассоциации искусственного интеллекта.

Автор более 100 печатных работ и 6 монографий.

Производственное издание

**Ойхман Ефим Григорьевич
Попов Эдуард Викторович**

**РЕИНЖИНИРИНГ БИЗНЕСА:
РЕИНЖИНИРИНГ ОРГАНИЗАЦИЙ
И ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ**

**Редакторы Л.Д.Григорьева, Л.А.Табакова
Художественный редактор Ю.И.Артюхов
Технический редактор Е.В.Кузьмина
Корректоры
Н.Б.Вторушина, Н.П.Сперанская
Художественное оформление Е.К.Самойлова**

ИБ № 3688

Лицензия ЛР № 010156 от 29.01.1997

Подписано в печать 04.04.97 Формат 60х88/16
Гарнитура "Таймс". Печать офсетная
Усл.печ л. 20,58. Уч.-изд.л. 21,31 Тираж 10000 экз.
Заказ 1227. "С" 034

Издательство "Финансы и статистика"
101000, Москва, ул. Покровка, 7
Телефон (095) 925-35-02, факс (095) 925-09-57

**Великолукская городская типография
Упринформпечати Псковской области
182100. Великие Луки, ул. Полиграфистов. 78/12**

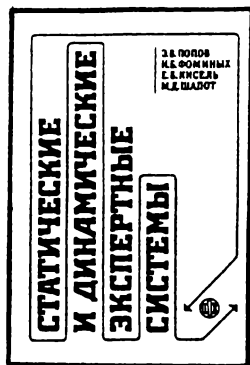
предлагает

**Э.В.Попов, И.Б.Фоминых,
Е.Б.Кисель, М.Д.Шапот**

"СТАТИЧЕСКИЕ И ДИНАМИЧЕСКИЕ ЭКСПЕРТНЫЕ СИСТЕМЫ"

17000 руб., 320 с.

Книга посвящена одной из центральных проблем искусственного интеллекта - проблеме построения экспертных систем (ЭС), или систем, основанных на знаниях. Рассмотрено современное состояние исследований в области искусственного интеллекта. Обоснована важность ЭС как подхода к разработке интегрированных прикладных систем, объединяющих технологии инженерии знаний и традиционного программирования. Приведена классификация ЭС и инструментальных средств. Описаны принципы, методы и средства построения ЭС. Особое внимание уделено новому классу экспертных систем - динамическим системам реального времени.

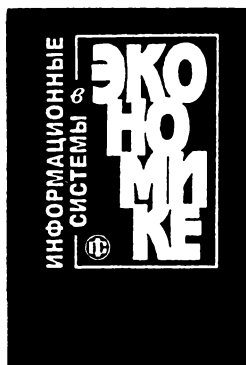


"ИНФОРМАЦИОННЫЕ СИСТЕМЫ В ЭКОНОМИКЕ"

Под. ред. В.В.Дика

15000 руб., 272 с.

Раскрываются основы применения информационных технологий как базы экономических информационных систем. Рассматриваются возможности создания и использования статистических, банковских, бухгалтерских информационных систем и информационных систем фондового рынка.



Книги Вы можете приобрести в издательстве по адресу:

101000, Москва, ул.Покровка, 7

(метро "Китай-город", выход на ул.Маросейка)

Тел.: (095) 925-35-02, 923-80-42 Факс (095) 925-09-57



**графическая,
объектно-ориентированная среда
для создания динамических
экспертных систем**



Gensym
Intelligent Real-Time Systems

Система G2, разработанная корпорацией Gensym, помогает компаниям активизировать самый ценный капитал - знания своих сотрудников. С ее помощью Вы можете легко разрабатывать и сопровождать интеллектуальные приложения реального времени. Используя G2, Вы сокращаете время и затраты на разработку. По сообщениям пользователей G2, применение этой системы дает почти десятикратный рост производительности по сравнению с традиционными методами программирования.

Классы задач, для которых предназначена G2:

- мониторинг в реальном времени
- моделирование
- системы управления верхнего уровня
- оптимизация
- диагностика неисправностей
- системы - советчики оператора
- планирование и составление расписаний
- системы проектирования

С момента выхода первой версии системы в 1988 г. в мире установлено более 5000 лицензий на продукты Gensym. Фирма обслуживает около 30 отраслей, от аэрокосмических исследований до производства пищевых продуктов. Список пользователей G2 выглядит как справочник Who-Is-Who в мировой промышленности: 40 из 50 самых крупных компаний мира используют G2.

Успех инструментального комплекса G2 объясняется открытостью, ориентацией на промышленные стандарты и поддержкой широкого спектра платформ: Sun, DEC, HP, IBM и ПЭВМ на базе процессоров Intel. Продукты фирмы Gensym имеют богатые интеграционные и сетевые возможности. Возможность включения G2 в уже существующие системы управления производством обеспечивается развитыми интерфейсами с базами данных (Oracle, Sybase, Informix, Rdb и все ODBC совместимые СУБД), программируемыми контроллерами ведущих фирм (Allen Bradley, GE-Fanuc, AEG Modicon), системами сбора данных (ABB, Fisher, Siemens, Yokogawa, Foxboro, ORSI), концентраторами данных (DEC BASEstar, Allen Bradley Pyramid Integrator, SETPOINT SETCIM).



Argussoft Co - представитель Gensym Corp. в России
129090, Россия, Москва, ул. Щепкина, д. 22, подъезд 3, оф. 42
Тел.: (095) 284-8229 / 288-2436 Факс: (095) 288-2085
E-mail: g2group@arguss.msk.su

ReThink



Gensym
Intelligent Reeth Think Systems

система для тех, кто желает:

- упростить анализ и моделирование компании,
- оценить экономические выгоды от предлагаемой реорганизации и аргументированно продемонстрировать их оппонентам
- и после этого ..., не выбрасывая на ветер затраты на создание модели, использовать ее для точной настройки хозяйственного механизма и контроля за ходом бизнес-процессов.

Одно из наиболее модных и динамично развивающихся направлений сегодня - реорганизация бизнес-процессов (Business Process Reengineering - BPR). Его цель - дать возможность пользователям увеличить эффективность функционирования их предприятиями. Gensym предлагает систему моделирования ReThink для реализации приложений в области BPR.

ReThink ориентирован не на программистов - системных аналитиков, а на обычных управленцев. Такой подход устраняет ошибки взаимного непонимания между заказчиком и исполнителем, поскольку в данном случае они объединены в одном лице. Еще одним положительным качеством системы является поддержка полного цикла разработки приложения, а не отдельных его этапов.

В системе ReThink бизнес-процессы представляются графически в виде пиктограмм, соединенных связями, отображающими последовательность выполнения и взаимодействие между существенными с точки зрения управленца задачами. Задачи могут объединяться в произвольном порядке для визуализации процессов на любом уровне детализации.

Для сравнения вариантов организации работ при различных внешних условиях введен механизм сценариев, каждый из которых представляет независимый блок управления моделированием дискретных событий.

Система ReThink, базируясь на G2, обеспечивает визуализацию функционирования модели на основе активной графики, архитектуру клиент-сервер и возможности соединения ReThink с "живыми" данными из СУБД и АСУТП, что позволяет в конечном счете превратить модель в рабочую версию системы организационного управления.



Argussoft Co - представитель Gensym Corp. в России
129090, Россия, Москва, ул. Щепкина, д. 22, подъезд 3, оф. 42
Тел.: (095) 284-8229 / 288-2436 Факс: (095) 288-2085
E-mail: g2group@arguss.msk.su