

# Jet Info

ИНФОРМАЦИОННЫЙ БЮЛЛЕТЕНЬ

№ 10 (137)/2004

## Методы и средства моделирования бизнес-процессов

КОРПОРАТИВНЫЕ  
СИСТЕМЫ



# Методы и средства моделирования бизнес-процессов (обзор)

А.М. Вендров

## СОДЕРЖАНИЕ

---

1. Введение.....	3
2. Основные понятия.....	3
3. Методы моделирования бизнес-процессов.....	5
3.1. Метод функционального моделирования SADT (IDEF0)	
3.2. Метод моделирования процессов IDEF3	
3.3. Моделирование потоков данных DFD	
3.4. Метод ARIS	
3.5. Метод Ericsson-Penker и образцы моделирования бизнес-процессов	
3.6. Метод моделирования, используемый в технологии Rational Unified Process	
4. Сравнительный анализ различных методов и инструментальных средств моделирования .....	28
5. Перспективные направления в моделировании бизнес-процессов .....	30
5.1. Деятельность консорциума Business Process Management Initiative (BPMI)	
5.2. Проект UEML	
5.3. Работы в рамках проекта OMG MDA	
Библиография.....	32

---

## 1. Введение

В современной практике моделирования управленческой и производственной деятельности для обозначения объектов моделирования принято использовать термин **«бизнес-процесс» (business-process)** [Репин-04]. В международном стандарте ISO 9000:2000 принят термин «процесс», однако в настоящее время эти термины можно считать синонимами.

Моделирование бизнес-процессов является важной составной частью проектов по реинжинирингу (реорганизации) бизнес-процессов и созданию крупномасштабных систем программного обеспечения (ПО) [Ойхман-97]. Отсутствие таких моделей является одной из главных причин неудач многих проектов. Однако даже наличие моделей не гарантирует успеха, поскольку существует ряд других серьезных причин, приводящих к неудачам:

- отсутствие заинтересованности со стороны высшего руководства организации;
- некорректная постановка целей проекта;
- недостаточная информированность персонала организации относительно целей и результатов проекта;
- непонимание сути и реальных возможностей используемых методов моделирования;
- отсутствие корпоративных стандартов описания и регламентации бизнес-процессов;
- неэффективное применение инструментов моделирования.

По оценкам, приведенным в [Репин-04], типовой сценарий развития событий в большинстве российских организаций заключается в следующем: ставятся «правильные» цели, инициируется проект, создается описание бизнес-процессов, осуществляются попытки проанализировать их и приступить к реорганизации. Именно на последнем этапе, когда необходимо получить определенные результаты, большинство организаций испытывает значительные трудности. Не получив быстрых, измеримых результатов, предвидя длительную и кропотливую работу, руководство организации сворачивает работу по проекту, и начинается поиск очередных «модных» подходов к управлению.

Сложившаяся ситуация еще более усугубляется следующими обстоятельствами:

- организации сталкиваются с проблемой выбора адекватных методов и инструментов моделирования, которая порождается их разнообразием и отсутствием единых стандартов;

- существующие методы и средства используют различные языки моделирования, терминологию, плохо совместимы друг с другом, дорогостоящи и трудоемки в использовании.

Эти обстоятельства обуславливают многочисленные проекты, предпринимаемые в настоящее время, целью которых является интеграция существующих методов и языков моделирования и создание единого методического и технологического базиса моделирования бизнес-процессов, а в более широком контексте — моделирования архитектуры предприятий (enterprise modeling) [BPMN-03, UEMML-02, BPDM-03].

## 2. Основные понятия

**Бизнес-процесс** определяется как логически завершенный набор взаимосвязанных и взаимодействующих видов деятельности, поддерживающий деятельность организации и реализующий ее политику, направленную на достижение поставленных целей. Стандарт ISO 9000:2000 определяет процесс как совокупность взаимосвязанных и взаимодействующих видов деятельности, преобразующих входы в выходы, представляющие ценность для потребителя. В этом определении под процессом можно понимать любую деятельность, использующую определенные ресурсы (финансовые, материальные, человеческие, информационные) для преобразования входных элементов в выходные. Процесс включает одну или более связанных между собой процедур или функций, которые совместно реализуют некоторую задачу (обычно в рамках организационной структуры). Он может выполняться в пределах одной организационной единицы, охватывать несколько единиц или даже несколько различных организаций.

Важным шагом структуризации деятельности любой организации являются выделение и классификация бизнес-процессов. По отношению к получению добавленной ценности продукта или услуги можно выделить следующие классы процессов:

- **основные процессы;**
- **обеспечивающие процессы.**

Основными бизнес-процессами являются процессы, добавляющие ценность. Они ориентированы на производство товаров или оказание услуг, составляющих основную деятельность организации и обеспечивающих получение дохода. Примерами таких процессов на предприятии являются процессы маркетинга, производства, поставки и сервисного обслуживания продукции.

Обеспечивающие бизнес-процессы не добавляют ценность продукта или услуги для потребителя, но увеличивают их стоимость. Они необходимы для деятельности предприятия и предназначены для поддержки выполнения основных бизнес-процессов. Такими процессами являются финансовое обеспечение деятельности, обеспечение кадрами, юридическое обеспечение, администрирование, обеспечение безопасности, поставка комплектующих материалов, ремонт и техническое обслуживание и т.д.

Бизнес-процессы можно также классифицировать по видам деятельности или составу работ (элементам процесса) [Репин-04]:

- планирование деятельности (например, планирование производства готовой продукции);
- осуществление деятельности — собственно выполнение работы (например, изготовление продукции);
- регистрация фактической информации по выполнению процесса (производственный, управленческий и бухгалтерский учет);
- контроль и анализ исполнения плана;
- принятие управленческих решений. Эти процессы охватывают весь комплекс функций управления на уровне каждого бизнес-процесса и системы в целом. Примерами таких процессов могут быть процессы стратегического, оперативного и текущего планирования, процессы формирования и выполнения управляющих воздействий. Процессы управления оказывают воздействие на все остальные процессы организации.

**Бизнес-модель** — это формализованное (графическое, табличное, текстовое, символическое) описание бизнес-процессов, отражающее реально существующую или предполагаемую деятельность предприятия.

В простейшем случае бизнес-модель может состоять из единственной диаграммы, однако на практике это вряд ли допустимо, поскольку бизнес-процессы, как правило, слишком сложны и многоаспектны. Модель таких про-

цессов включает следующие компоненты [Eriksson-2000]:

- Представления. Каждое представление отражает определенный аспект бизнес-процессов. Представление — это абстракция, отражающая конкретную точку зрения и скрывающая детали, несущественные для данной точки зрения.
- Диаграммы. Каждое представление состоит из ряда диаграмм различных типов, отражающих структурные и динамические аспекты бизнес-процессов.
- Объекты и процессы. Объекты представляют ресурсы, используемые в процессах (финансовые, материальные, человеческие, информационные).

Цели моделирования бизнес-процессов обычно формулируются следующим образом:

- обеспечить понимание структуры организации и динамики происходящих в ней процессов;
- обеспечить понимание текущих проблем организации и возможностей их решения;
- убедиться, что заказчики, пользователи и разработчики одинаково понимают цели и задачи организации;
- создать базу для формирования требований к ПО, автоматизирующему бизнес-процессы организации.

Основная область применения бизнес-моделей — это реинжиниринг бизнес-процессов. При этом предполагается построение моделей текущей и перспективной деятельности, а также плана и программы перехода из первого состояния во второе. Любое современное предприятие является сложной системой, его деятельность включает в себя исполнение десятков тысяч взаимовлияющих функций и операций. Человек не в состоянии понимать, как такая система функционирует в деталях — это выходит за границы его возможностей. Поэтому главная идея создания так называемых моделей «AS-IS» (как есть) и «AS-TO-BE» (как должно быть) — понять, что делает (будет делать) рассматриваемое предприятие и как оно функционирует (будет функционировать) для достижения своих целей.

Назначением будущих систем ПО является, в первую очередь, решение проблем бизнеса посредством современных информационных технологий. Требования к ПО формируются на основе бизнес-модели, а критерии проектирования систем прежде всего основываются на наиболее полном их удовлетворении.



Следует отметить, что модели бизнес-процессов являются не просто промежуточным результатом, используемым консультантом для выработки каких-либо рекомендаций и заключений. Они представляют собой самостоятельный результат, имеющий большое практическое значение, которое следует из целей их построения.

Модель бизнес-процесса должна давать ответы на вопросы:

1. Какие процедуры (функции, работы) необходимо выполнить для получения заданного конечного результата?
2. В какой последовательности выполняются эти процедуры?
3. Какие механизмы контроля и управления существуют в рамках рассматриваемого бизнес-процесса?
4. Кто выполняет процедуры процесса?
5. Какие входящие документы/информацию использует каждая процедура процесса?
6. Какие исходящие документы/информацию генерирует процедура процесса?
7. Какие ресурсы необходимы для выполнения каждой процедуры процесса?
8. Какая документация/условия регламентирует выполнение процедуры?
9. Какие параметры характеризуют выполнение процедур и процесса в целом?

Важным элементом модели бизнес-процессов являются **бизнес-правила** или правила предметной области. Типичными бизнес-правилами являются корпоративная политика и государственные законы. Бизнес-правила обычно формулируются в специальном документе и могут отражаться в моделях. Для организации бизнес-правил предлагается множество различных схем классификации. Наиболее полной можно считать следующую классификацию бизнес-правил (в скобках приведены примеры правил для гипотетической системы обработки заказов в торговой компании):

- **Факты** — достоверные утверждения о бизнес-процессах, называемые также инвариантами (оплачивается доставка каждого заказа; со стоимости доставки налог с продаж не берется).
- **Правила-ограничения** — определяют различные ограничения на выполняемые операции:
  - **Управляющие воздействия и реакции на воздействия** (когда заказ отменен и еще не доставлен, то его обработка завершается).
  - **Операционные ограничения** — условия и постусловия (доставить заказ клиенту только при наличии адреса доставки).

- **Структурные ограничения** (заказ включает по крайней мере один продукт).
- **Активаторы операций** — правила, при определенных условиях приводящие к выполнению каких-либо действий (если срок хранения товара на складе истек, об этом надо уведомить ответственное лицо).
- **Правила вывода:**
  - **Правила-следствия** — правила, устанавливающие новые факты на основе достоверности определенных условий (клиент получает положительный статус только при условии оплаты счетов в течение 30 дней).
  - **Вычислительные правила** — различные вычисления, выполняемые с использованием математических формул и алгоритмов ( $\text{цена нетто} = \text{цена продукта} \cdot (1 + \text{процент налога} / 100)$ ).

Для моделирования бизнес-процессов необходимо использовать определенную методику, которая включает:

- описание методов моделирования — способов представления реальных объектов предприятия при помощи объектов модели;
- процедуру — последовательность шагов по сбору информации, ее обработке и представлению в виде моделей (диаграмм и документов).

Методика может существовать как самостоятельный продукт (например, метод Ericsson-Penker [Eriksson-2000]) или входить в состав комплексной технологии создания ПО (например, метод моделирования бизнес-процессов в технологии Rational Unified Process).

### 3. Методы моделирования бизнес-процессов

Для моделирования бизнес-процессов используется несколько различных методов, основой которых являются как структурный, так и объектно-ориентированный подходы к моделированию. Однако деление самих методов на структурные и объектные является достаточно условным, поскольку наиболее развитые методы

используют элементы обоих подходов. К числу наиболее распространенных методов относятся:

- метод функционального моделирования SADT (IDEF0);
- метод моделирования процессов IDEF3;
- моделирование потоков данных DFD;
- метод ARIS;
- метод Ericsson-Penker;
- метод моделирования, используемый в технологии Rational Unified Process.

### 3.1. Метод функционального моделирования SADT (IDEF0)

Метод SADT (Structured Analysis and Design Technique) [Марка-93, Черемных-01, Репин-04] считается классическим методом процессного подхода к управлению. Основной принцип процессного подхода заключается в структурировании деятельности организации в соответствии с ее бизнес-процессами, а не организационно-штатной структурой. Именно бизнес-процессы, формирующие значимый для потребителя результат, представляют ценность, и именно их улучшением предстоит в дальнейшем заниматься. Модель, основанная на организационно-штатной структуре, может продемонстрировать лишь хаос, царящий в организации (о котором в принципе руководству и так известно, иначе оно бы не инициировало соответствующие работы), на ее основе можно только внести предложения об изменении этой структуры. С другой стороны, модель, основанная на бизнес-процессах, содержит в себе и организационно-штатную структуру предприятия.

В соответствии с этим принципом бизнес-модель должна выглядеть следующим образом:

1. Верхний уровень модели должен отражать только контекст системы – взаимодействие моделируемого единственным контекстным процессом предприятия с внешним миром.
2. На втором уровне модели должны быть отражены основные виды деятельности (тематически сгруппированные бизнес-процессы) предприятия и их взаимосвязи. В случае большого их количества некоторые из них можно вынести на третий уровень модели. Но в любом случае под виды деятельности необходимо отводить не более двух уровней модели.
3. Дальнейшая детализация бизнес-процессов осуществляется посредством бизнес-функций – совокупностей операций, сгруппированных по определенным признакам. Бизнес-функции детализируются с помощью элементарных бизнес-операций.

4. Описание элементарной бизнес-операции осуществляется посредством задания алгоритма ее выполнения.

Метод SADT разработан Дугласом Россом (SoftTech, Inc.) в 1969 г. для моделирования искусственных систем средней сложности.

Данный метод успешно использовался в военных, промышленных и коммерческих организациях США для решения широкого круга задач, таких как долгосрочное и стратегическое планирование, автоматизированное производство и проектирование, разработка ПО для оборонных систем, управление финансами и материально-техническим снабжением и др. Метод SADT поддерживается Министерством обороны США, которое было инициатором разработки семейства стандартов IDEF (Icam DEFinition), являющегося основной частью программы ICAM (интегрированная компьютеризация производства), проводимой по инициативе ВВС США. Метод SADT реализован в одном из стандартов этого семейства – IDEF0, который был утвержден в качестве федерального стандарта США в 1993 г., его подробные спецификации можно найти на сайте <http://www.idef.com>. Существует также российская версия данного стандарта [РД-2000]. Вместе со стандартом IDEF0 обычно используются стандарт моделирования процессов IDEF3 и стандарт моделирования данных IDEF1X.

Метод SADT представляет собой совокупность правил и процедур, предназначенных для построения функциональной модели объекта какой-либо предметной области. Функциональная модель SADT отображает функциональную структуру объекта, т.е. производимые им действия и связи между этими действиями. Основные элементы этого метода основываются на следующих концепциях:

- Графическое представление блочного моделирования. Графика блоков и дуг SADT-диаграммы отображает функцию в виде блока, а интерфейсы входа/выхода представляются дугами, соответственно входящими в блок и выходящими из него. Взаимодействие блоков друг с другом описывается посредством интерфейсных дуг, выражающих «ограничения», которые, в свою очередь, определяют когда и каким образом функции выполняются и управляются.
- Строгость и точность. Выполнение правил SADT требует достаточной строгости и точности, не накладывая в то же время чрезмерных ограничений на действия аналити-

ка. Правила SADT включают: ограничение количества блоков на каждом уровне декомпозиции (правило 3-6 блоков – ограничение мощности краткосрочной памяти человека), связность диаграмм (номера блоков), уникальность меток и наименований (отсутствие повторяющихся имен), синтаксические правила для графики (блоков и дуг), разделение входов и управлений (правило определения роли данных).

- Отделение организации от функции, т.е. исключение влияния административной структуры организации на функциональную модель.

Метод SADT может использоваться для моделирования самых разнообразных процессов и систем. В существующих системах метод SADT может быть использован для анализа функций, выполняемых системой, и указания механизмов, посредством которых они осуществляются.

### 3.1.1. Состав функциональной модели

Результатом применения метода SADT является модель, которая состоит из диаграмм, фрагментов текстов и глоссария, имеющих ссылки друг на друга. Диаграммы – главные компоненты модели, все функции организации и интерфейсы на них представлены как блоки и дуги соответственно. Место соединения дуги с блоком определяет тип интерфейса. Управляющая информация входит в блок сверху, в то время как входная информация, которая подвергается обработке, показана с левой стороны блока, а результаты (выход) показаны с правой стороны. Механизм (человек или автоматизированная система), который осуществляет операцию, представляется дугой, входящей в блок снизу (рис. 1).

Одной из наиболее важных особенностей метода SADT является постепенное введение все больших уровней детализации по мере создания диаграмм, отображающих модель.

На рис. 2, где приведены четыре диаграммы и их взаимосвязи, показана структура SADT-модели. Каждый компонент модели может быть декомпозирован на другой диаграмме. Каждая диаграмма иллюстрирует «внутреннее строение» блока на родительской диаграмме.

Построение SADT-модели заключается в выполнении следующих действий:

- сбор информации об объекте, определение его границ;
- определение цели и точки зрения модели;
- построение, обобщение и декомпозиция диаграмм;

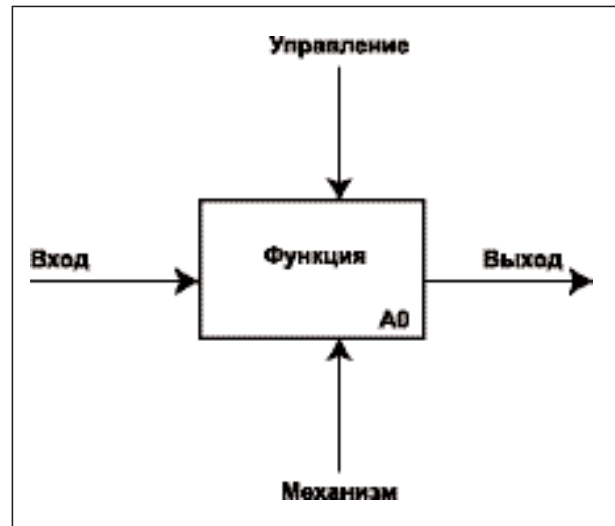


Рис. 1 Функциональный блок и интерфейсные дуги

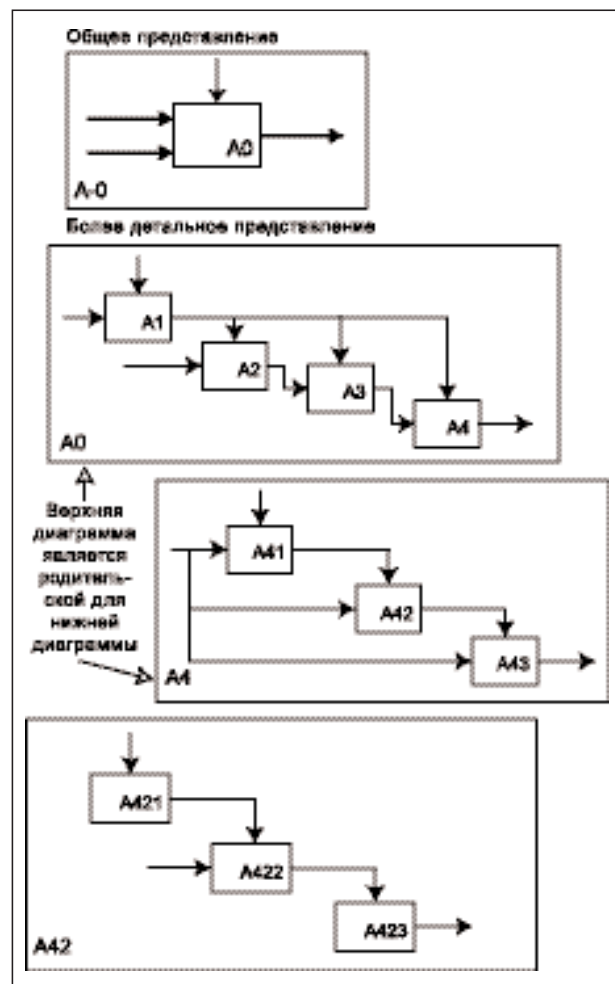


Рис. 2 Структура SADT-модели. Декомпозиция диаграмм

- критическая оценка, рецензирование и комментирование.

Построение диаграмм начинается с представления всей системы в виде простейшего компонента — одного блока и дуг, изображающих интерфейсы с функциями вне системы. Поскольку единственный блок отражает систему как единое целое, имя, указанное в блоке, является общим. Это верно и для интерфейсных дуг — они также соответствуют полному набору внешних интерфейсов системы в целом.

Затем блок, который представляет систему в качестве единого модуля, детализируется на другой диаграмме с помощью нескольких блоков, соединенных интерфейсными дугами. Эти блоки определяют основные подфункции исходной функции. Данная декомпозиция выявляет полный набор подфункций, каждая из которых показана как блок, границы которого определены интерфейсными дугами. Каждая из этих подфункций может быть декомпозирована подобным образом в целях большей детализации.

Во всех случаях каждая подфункция может содержать только те элементы, которые входят в исходную функцию. Кроме того, модель не может опустить какие-либо элементы, т.е., как уже отмечалось, родительский блок и его интерфейсы обеспечивают контекст. К нему нельзя ничего добавить, и из него не может быть ничего удалено.

Модель SADT представляет собой серию диаграмм с сопроводительной документацией, разбивающих сложный объект на составные части, которые изображены в виде блоков. Детали каждого из основных блоков показаны в виде блоков на других диаграммах. Каждая детальная диаграмма является декомпозицией блока из диаграммы предыдущего уровня. На каждом шаге декомпозиции диаграмма предыдущего уровня называется родительской для более детальной диаграммы.

На SADT-диаграммах не указаны явно ни последовательность, ни время. Обратные связи, итерации, продолжающиеся процессы и перекрывающиеся (по времени) функции могут быть изображены с помощью дуг. Обратные связи могут выступать в виде комментариев, замечаний, исправлений и т.д.

### 3.1.2. Стратегии декомпозиции

При построении иерархии диаграмм используются следующие стратегии декомпозиции:

- *Функциональная декомпозиция* — декомпозиция в соответствии с функциями, которые выполняют люди или организация. Может оказаться полезной стратегией для создания системы описаний, фиксирующей взаимодействие между людьми в процессе

их работы. Очень часто, однако, взаимосвязи между функциями весьма многочисленны и сложны, поэтому рекомендуется использовать эту стратегию только в начале работы над моделью системы.

- *Декомпозиция в соответствии с известными стабильными подсистемами* — приводит к созданию набора моделей, по одной модели на каждую подсистему или важный компонент. Затем для описания всей системы должна быть построена составная модель, объединяющая все отдельные модели. Рекомендуется использовать разложение на подсистемы, только когда разделение на основные части системы не меняется. Нестабильность границ подсистем быстро обесценит как отдельные модели, так и их объединение.
- *Декомпозиция по физическому процессу* — выделение функциональных стадий, этапов завершения или шагов выполнения. Хотя эта стратегия полезна при описании существующих процессов (таких, например, как работа промышленного предприятия), результатом ее часто может стать слишком последовательное описание системы, которое не будет в полной мере учитывать ограничения, диктуемые функциями друг другу. При этом может оказаться скрытой последовательность управления. Эта стратегия рекомендуется только если целью модели является описание физического процесса как такового или только в крайнем случае, когда неясно, как действовать.

Одна из наиболее частых проблем, возникающих в процессе построения SADT-моделей, — когда же следует завершить построение конкретной модели? На этот вопрос не всегда легко ответить, хотя существуют некоторые эвристики для определения разумной степени полноты. Здесь представлены правила, которыми пользуются опытные аналитики для определения момента завершения моделирования. Они носят характер рекомендаций. Только длительная практика позволит приобрести знания, необходимые для принятия правильного решения об окончании моделирования.

Рекомендуется прекращать моделирование, когда уровень детализации модели удовлетворяет ее цель. Опыт показал, что для отдельной модели, которая создается независимо от какой-либо другой модели, декомпозиция одного из ее блоков должна прекращаться, если:



- Блок содержит достаточно деталей. Одна из типичных ситуаций, встречающихся в конце моделирования — это блок, который описывает систему с нужным уровнем подробности. Проверить достаточность деталей обычно совсем легко, необходимо просто спросить себя, отвечает ли блок на все или на часть вопросов, составляющих цель модели. Если блок помогает ответить на один или более вопросов, то дальнейшая декомпозиция может не понадобиться.
- Необходимо изменить уровень абстракции, чтобы достичь большей детализации, блока. Блоки подвергаются декомпозиции, если они недостаточно детализированы для удовлетворения цели модели. Но иногда при декомпозиции блока выясняется, что диаграмма начинает описывать, как функционирует блок, вместо описания того, что блок делает. В этом случае происходит изменение уровня абстракции — изменение сути того, что должна представлять модель (т.е. изменение способа описания системы). В SADT изменение уровня абстракции часто означает выход за пределы цели модели и, следовательно, это указывает на прекращение декомпозиции.
- Необходимо изменить точку зрения, чтобы детализировать блок. Изменение точки зрения происходит примерно так же, как изменение уровня абстракции. Это чаще всего характерно для ситуаций, когда точку зрения модели нельзя использовать для декомпозиции конкретного блока, т. е. этот блок можно декомпонировать, только если посмотреть на него с другой позиции. Об этом может свидетельствовать заметное изменение терминологии.
- Блок очень похож на другой блок той же модели или на блок другой модели. Иногда встречается блок, чрезвычайно похожий на другой блок модели. Два блока похожи, если они выполняют примерно одну и ту же функцию и имеют почти одинаковые по типу и количеству входы, управления и выходы. Если второй блок уже декомпозирован, то разумно отложить декомпозицию и тщательно сравнить два блока. Если нужны ничтожные изменения для совпадения первого блока со вторым, то внесение этих изменений сократит усилия на декомпозицию и улучшит модульность модели (т.е. сходные функции уточняются согласованным образом).
- Блок представляет тривиальную функцию. Тривиальная функция — это такая функция, понимание которой не требует никаких объ-

яснений. В этом случае очевидна целесообразность отказа от декомпозиции, потому что роль SADT заключается в превращении сложного вопроса в понятный, а не в педантичной разработке очевидных деталей. В таких случаях декомпозиция определенных блоков может принести больше вреда, чем пользы. Тривиальные функции лучше всего описываются небольшим объемом текста. Следует заметить, что «тривиальный» не означает «бесполезный». Тривиальные функции выполняют очень важную роль, поясняя работу более сложных функций, а иногда и соединяя вместе основные подсистемы. Поэтому при анализе не следует пропускать тривиальные функции. Наоборот, их существование должно быть зафиксировано и они должны быть детализированы, как и любые другие функции. Однако следует предостеречь от больших затрат времени на анализ тривиальных функций системы. Усиленное внимание к мелочам может привести к созданию модели, которой будет не хватать абстракции, что сделает ее трудной для понимания и использования.

Общее число уровней в модели (включая контекстный) не должно превышать 5-6. Практика показывает, что этого вполне достаточно для построения полной функциональной модели современного предприятия любой отрасли.

Метод SADT в наибольшей степени подходит для описания процессов верхнего уровня управления. Его основные преимущества заключаются в следующем [Репин-04]:

- полнота описания бизнес-процесса (управление, информационные и материальные потоки, обратные связи);
- комплексность декомпозиции;
- возможность агрегирования и детализации потоков данных и информации (разделение и слияние дуг);
- наличие жестких требований, обеспечивающих получение моделей стандартного вида;
- простота документирования процессов;
- соответствие подхода к описанию процессов стандарту ISO 9000:2000.

В то же время метод SADT обладает рядом недостатков:

- сложность восприятия (большое количество дуг на диаграммах);
- большое количество уровней декомпозиции;

- трудность увязки нескольких процессов, представленных в различных моделях одной и той же организации.

### 3.2. Метод моделирования процессов IDEF3

Метод моделирования IDEF3 [Черемных-01, Репин-04], являющийся частью семейства стандартов IDEF, был разработан в конце 1980-х годов для закрытого проекта ВВС США. Этот метод предназначен для моделирования последовательности выполнения действий и взаимозависимости между ними в рамках процессов. Хотя IDEF3 и не достиг статуса федерального стандарта США, он приобрел широкое распространение среди системных аналитиков как дополнение к методу функционального моделирования IDEF0 (модели IDEF3 могут использоваться для детализации функциональных блоков IDEF0, не имеющих диаграмм декомпозиции).

Основой модели IDEF3 служит так называемый сценарий процесса, который выделяет последовательность действий и подпроцессов анализируемой системы.

Как и в методе IDEF0, основной единицей модели IDEF3 является диаграмма. Другой важный компонент модели — действие, или в терминах IDEF3 «единица работы» (Unit of Work). Диаграммы IDEF3 отображают действие в виде прямоугольника. Действия именуются с использованием глаголов или отглагольных существительных, каждому из действий присваивается уникальный идентификационный номер. Этот номер не используется вновь даже в том случае, если в процессе построения модели действие удаляется. В диаграммах IDEF3 номер действия обычно предваряется номером его родителя (рис. 3).

Существенные взаимоотношения между действиями изображаются с помощью связей. Все связи в IDEF3 являются однонаправленными, и хотя стрелка может начинаться или заканчиваться на любой стороне блока, обозначаю-



Рис. 3 Изображение и нумерация действия в диаграмме IDEF3

щего действие, диаграммы IDEF3 обычно организованы слева направо таким образом, что стрелки начинаются на правой и заканчиваются на левой стороне блоков. В табл. 1 приведены три возможных типа связей.

Связь типа «временное предшествование» показывает, что исходное действие должно полностью завершиться, прежде чем начнется выполнение конечного действия.

Связь типа «объектный поток» используется в том случае, когда некоторый объект, являющийся результатом выполнения исходного действия, необходим для выполнения конечного действия. Обозначение такой связи отличается от связи временного предшествования двойной стрелкой. Наименования потоковых связей должны четко идентифицировать объект, который передается с их помощью. Временная семантика объектных связей аналогична связям предшествования, это означает, что порождающее объектную связь исходное действие должно завершиться, прежде чем конечное действие может начать выполняться.

Связь типа «нечеткое отношение» используется для выделения отношений между действиями, которые невозможно описать с использованием связей предшествования или объектных связей. Значение каждой такой связи должно быть определено, поскольку связи типа «нечеткое отношение» сами по себе не предполагают никаких ограничений. Одно из применений нечетких отношений — отображе-

Изображение	Название	Назначение
	Временное предшествование (Temporal precedence)	Исходное действие должно завершиться, прежде чем конечное действие сможет начаться
	Объектный поток (Object flow)	Выход исходного действия является входом конечного действия (исходное действие должно завершиться, прежде чем конечное действие сможет начаться)
	Нечеткое отношение (Relationship)	Вид взаимодействия между исходным и конечным действиями задается аналитиком отдельно для каждого случая использования такого отношения

Табл. 1 Типы связей IDEF3

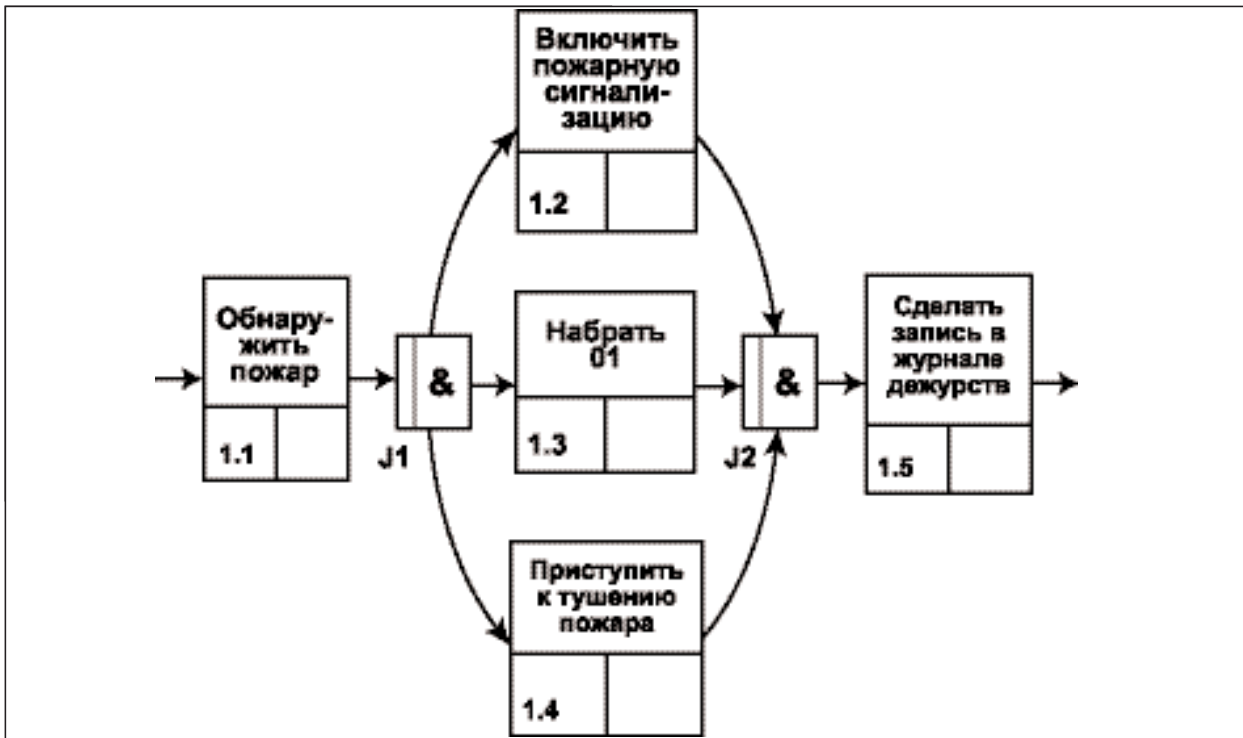


Рис. 4 Соединения «и»

ние взаимоотношений между параллельно выполняющимися действиями.

Завершение одного действия может инициировать начало выполнения сразу нескольких других действий или, наоборот, определенное действие может требовать завершения нескольких других действий до начала своего выполнения. Соединения разбивают или соединяют внутренние потоки и используются для изображения ветвления процесса:

- разворачивающие соединения используются для разбиения потока. Завершение одного действия вызывает начало выполнения нескольких других;
- сворачивающие соединения объединяют потоки. Завершение одного или нескольких действий вызывает начало выполнения другого действия.

Соединения «и» инициируют выполнение конечных действий. Все действия, присоединенные к сворачивающему соединению «и», должны завершиться, прежде чем начнется выполнение следующего действия. На рис. 4 после обнаружения пожара инициируются включение пожарной сигнализации, вызов пожарной охраны, и начинается тушение пожара. Запись в журнал производится только тогда, когда все три перечисленных действия завершены.

Соединение «исключающее «или»» означает, что вне зависимости от количества дейст-

вий, связанных со сворачивающим или разворачивающим соединением, инициировано будет только одно из них, и поэтому только оно будет завершено перед тем, как любое действие, следующее за сворачивающим соединением, сможет начаться. Если правила активации соединения известны, они обязательно должны быть документированы либо в его описании, либо пометкой стрелок, исходящих из разворачивающего соединения. На рис. 5 соединение «исключающее «или»» используется для отображения того факта, что студент не может одновременно быть направлен на лекции по двум разным курсам.

Соединение «или» предназначено для описания ситуаций, которые не могут быть описаны двумя предыдущими типами соединений. Аналогично связи нечеткого отношения соединение «или» в основном определяется и описывается непосредственно аналитиком. На рис. 6 соединение J2 может активизировать проверку данных чека и/или проверку суммы наличных. Проверка чека инициируется, если покупатель желает расплатиться чеком, проверка суммы наличных — при оплате наличными. И то, и другое действие инициируются при частичной оплате как чеком, так и наличными.

В рассмотренных примерах все действия выполнялись асинхронно, т.е. они не инициировались одновременно. Однако существуют случаи, когда время начала или окончания параллельно выполняемых действий должно быть

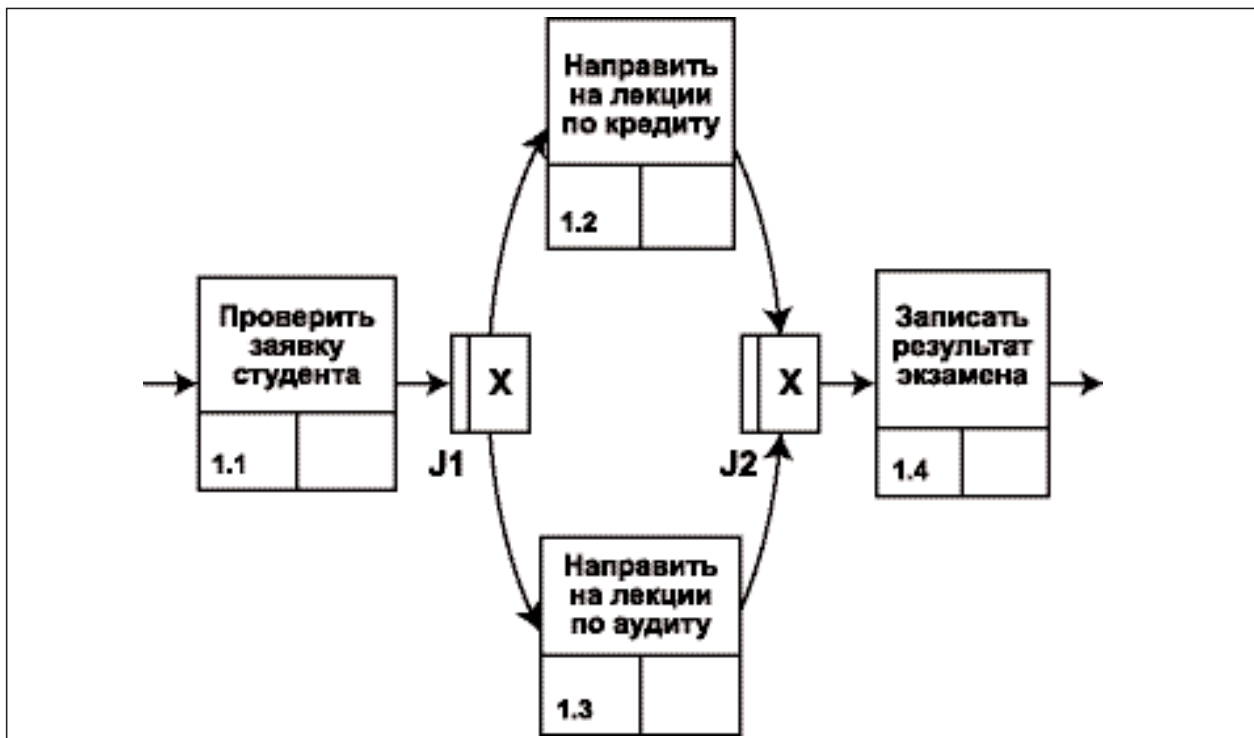


Рис. 5 Соединение «исключающее «или»»

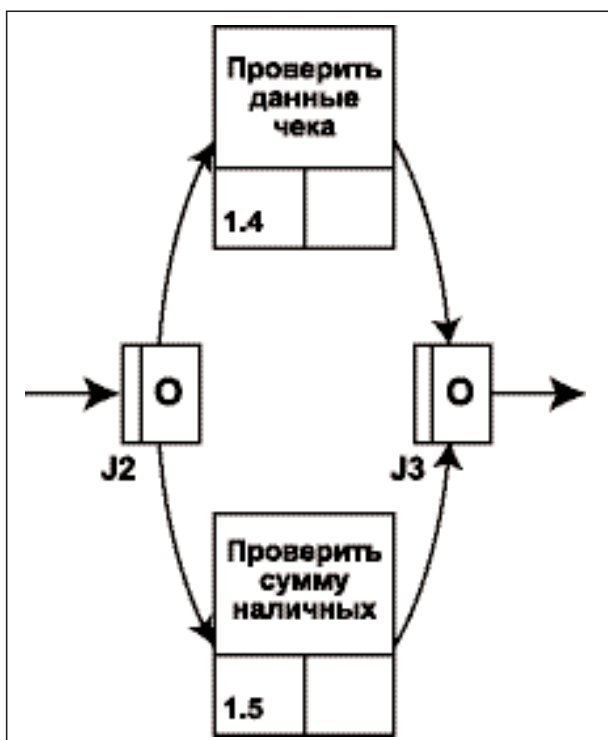


Рис. 6 Соединения «или»

одинаковым, т.е. действия должны выполняться синхронно. Для моделирования такого поведения системы используются различные виды синхронных соединений, которые обозначают-

ся двумя двойными вертикальными линиями внутри прямоугольника.

Все соединения на диаграммах должны быть парными, из чего следует, что любое разворачивающее соединение имеет парное себе сворачивающее. Однако типы соединений не обязательно должны совпадать.

Соединения могут комбинироваться для создания более сложных ветвлений. Комбинации соединений следует использовать с осторожностью, поскольку перегруженные ветвлениями диаграммы могут оказаться сложными для восприятия.

Действия в IDEF3 могут быть декомпозированы или разложены на составляющие для более детального анализа. Метод IDEF3 позволяет декомпозировать действие несколько раз, что обеспечивает документирование альтернативных потоков процесса в одной модели.

### 3.3. Моделирование потоков данных

**Диаграммы потоков данных** (Data Flow Diagrams – DFD) [Калашян-03] представляют собой иерархию функциональных процессов, связанных потоками данных. Цель такого представления – продемонстрировать, как каждый процесс преобразует свои входные данные в выходные, а также выявить отношения между этими процессами.



Для построения DFD традиционно используются две различные нотации, соответствующие методам Йордона-ДеМарко и Гейна-Сэрсона. Эти нотации незначительно отличаются друг от друга графическим изображением символов (далее в примерах используется нотация Гейна-Сэрсона).

В соответствии с данным методом модель системы определяется как иерархия диаграмм потоков данных, описывающих асинхронный процесс преобразования информации от ее ввода в систему до выдачи потребителю. Источники информации (внешние сущности) порождают информационные потоки (потоки данных), переносящие информацию к подсистемам или процессам. Те, в свою очередь, преобразуют информацию и порождают новые потоки, которые переносят информацию к другим процессам или подсистемам, накопителям данных или внешним сущностям – потребителям информации.

Диаграммы верхних уровней иерархии (контекстные диаграммы) определяют основные процессы или подсистемы с внешними входами и выходами. Они детализируются при помощи диаграмм нижнего уровня. Такая декомпозиция продолжается, создавая многоуровневую иерархию диаграмм, до тех пор, пока не будет достигнут уровень декомпозиции, на котором детализировать процессы далее не имеет смысла.

### 3.3.1. Состав диаграмм потоков данных

Основными компонентами диаграмм потоков данных являются:

- внешние сущности;
- системы и подсистемы;
- процессы;
- накопители данных;
- потоки данных.

Внешняя сущность представляет собой материальный объект или физическое лицо, являющиеся источником или приемником информации, например, заказчики, персонал, поставщики, клиенты, склад. Определение некоторого объекта или системы в качестве внешней сущности указывает на то, что она находится за пределами границ анализируемой системы. В процессе анализа некоторые внешние сущности могут быть перенесены внутрь диаграммы анализируемой системы, если это необходимо, или, наоборот, часть процессов может быть вынесена за пределы диаграммы и представлена как внешняя сущность.

**Внешняя сущность** обозначается квадратом (рис. 7), расположенным над диаграммой и бросающим на нее тень для того, чтобы можно было выделить этот символ среди других обозначений.



Рис. 7 Графическое изображение внешней сущности

При построении модели сложной системы она может быть представлена в самом общем виде на так называемой контекстной диаграмме в виде одной системы как единого целого, либо может быть декомпозирована на ряд подсистем.

Подсистема (или система) на контекстной диаграмме изображается так, как она представлена на рис. 8.

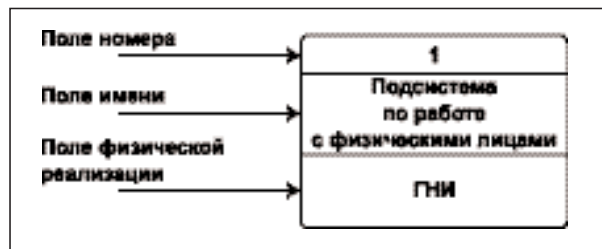


Рис. 8 Подсистема по работе с физическими лицами (ГНИ – Государственная налоговая инспекция)

Номер подсистемы служит для ее идентификации. В поле имени вводится наименование подсистемы в виде предложения с подлежащим и соответствующими определениями и дополнениями.

**Процесс** представляет собой преобразование входных потоков данных в выходные в соответствии с определенным алгоритмом. Физически процесс может быть реализован различными способами: это может быть подразделение организации (отдел), выполняющее обработку входных документов и выпуск отчетов, программа, аппаратно реализованное логическое устройство и т.д.

Процесс на диаграмме потоков данных изображается, как показано на рис. 9.



Рис. 9 Графическое изображение процесса

Номер процесса служит для его идентификации. В поле имени вводится наименование процесса в виде предложения с активным недвусмысленным глаголом в неопределенной форме (вычислить, рассчитать, проверить, определить, создать, получить), за которым следуют существительные в винительном падеже, например: «Ввести сведения о налогоплательщиках», «Выдать информацию о текущих расходах», «Проверить поступление денег».

Информация в поле физической реализации показывает, какое подразделение организации, программа или аппаратное устройство выполняет данный процесс.

**Накопитель данных** — это абстрактное устройство для хранения информации, которую можно в любой момент поместить в накопитель и через некоторое время извлечь, причем способы помещения и извлечения могут быть любыми.

Накопитель данных может быть реализован физически в виде микрофиши, ящика в картотеке, таблицы в оперативной памяти, файла на магнитном носителе и т.д. Накопитель данных на диаграмме потоков данных изображается, как показано на рис. 10.

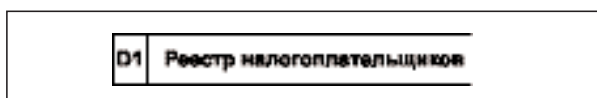


Рис. 10 Графическое изображение накопителя данных

Накопитель данных идентифицируется буквой «D» и произвольным числом. Имя накопителя выбирается из соображения наибольшей информативности для проектировщика.

Накопитель данных в общем случае является прообразом будущей базы данных, и описание хранящихся в нем данных должно соответствовать модели данных.

**Поток данных** определяет информацию, передаваемую через некоторое соединение от источника к приемнику. Реальный поток данных может быть информацией, передаваемой по кабелю между двумя устройствами, пересы-

лаемыми по почте письмами, магнитными лентами или дискетами, переносимыми с одного компьютера на другой и т.д.

Поток данных на диаграмме изображается линией, оканчивающейся стрелкой, которая показывает направление потока (рис. 11). Каждый поток данных имеет имя, отражающее его содержание.

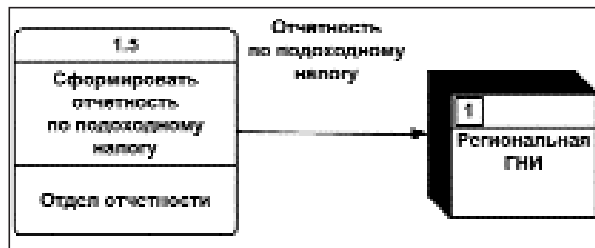


Рис. 11 Поток данных

### 3.3.2. Построение иерархии диаграмм потоков данных

Главная цель построения иерархии DFD заключается в том, чтобы сделать описание системы ясным и понятным на каждом уровне детализации, а также разбить его на части с точно определенными отношениями между ними. Для достижения этого целесообразно пользоваться следующими рекомендациями:

- Размещать на каждой диаграмме от 3 до 6-7 процессов (аналогично SADT). Верхняя граница соответствует человеческим возможностям одновременного восприятия и понимания структуры сложной системы с множеством внутренних связей, нижняя граница выбрана по соображениям здравого смысла: нет необходимости детализировать процесс диаграммой, содержащей всего один или два процесса.
- Не загромождать диаграммы несущественными на данном уровне деталями.
- Декомпозицию потоков данных осуществлять параллельно с декомпозицией процессов. Эти две работы должны выполняться одновременно, а не одна после завершения другой.
- Выбирать ясные, отражающие суть дела имена процессов и потоков, при этом стараться не использовать аббревиатуры.

Первым шагом при построении иерархии DFD является построение контекстных диаграмм. Обычно при проектировании относительно простых систем строится единственная контекстная диаграмма со звездообразной топологией, в центре которой находится так называемая

мый главный процесс, соединенный с приемниками и источниками информации, посредством которых с системой взаимодействуют пользователи и другие внешние системы. Перед построением контекстной DFD необходимо проанализировать внешние события (внешние сущности), оказывающие влияние на функционирование системы. Количество потоков на контекстной диаграмме должно быть по возможности небольшим, поскольку каждый из них может быть в дальнейшем разбит на несколько потоков на следующих уровнях диаграммы.

Для проверки контекстной диаграммы можно составить список событий. Список событий должен состоять из описаний действий внешних сущностей (событий) и соответствующих реакций системы на события. Каждое событие должно соответствовать одному или более потокам данных: входные потоки интерпретируются как воздействия, а выходные потоки — как реакции системы на входные потоки.

Для сложных систем (признаками сложности могут быть наличие большого количества внешних сущностей (десять и более), распределенная природа системы или ее многофункциональность) строится иерархия контекстных диаграмм. При этом контекстная диаграмма верхнего уровня содержит не единственный главный процесс, а набор подсистем, соединенных потоками данных. Контекстные диаграммы следующего уровня детализируют контекст и структуру подсистем.

Для каждой подсистемы, присутствующей на контекстных диаграммах, выполняется ее детализация при помощи DFD. Это можно сделать путем построения диаграммы для каждого события. Каждое событие представляется в виде процесса с соответствующими входными и выходными потоками, накопителями данных, внешними сущностями и ссылки на другие процессы для описания связей между этим процессом и его окружением. Затем все построенные диаграммы сводятся в одну диаграмму нулевого уровня.

Каждый процесс на DFD, в свою очередь, может быть детализирован при помощи DFD или (если процесс элементарный) спецификации. Спецификация процесса должна формулировать его основные функции таким образом, чтобы в дальнейшем специалист, выполняющий реализацию проекта, смог выполнить их или разработать соответствующую программу.

Спецификация является конечной вершиной иерархии DFD. Решение о завершении детализации процесса и использовании специфика-

ции принимается аналитиком исходя из следующих критериев:

- наличия у процесса относительно небольшого количества входных и выходных потоков данных (2-3 потока);
- возможности описания преобразования данных процессов в виде последовательного алгоритма;
- выполнения процессом единственной логической функции преобразования входной информации в выходную;
- возможности описания логики процесса при помощи спецификации небольшого объема (не более 20-30 строк).

Спецификации представляют собой описания алгоритмов задач, выполняемых процессами. Они содержат номер и/или имя процесса, списки входных и выходных данных и тело (описание) процесса, являющееся спецификацией алгоритма или операции, трансформирующей входные потоки данных в выходные. Языки спецификаций могут варьироваться от структурированного естественного языка или псевдокода до визуальных языков моделирования.

Структурированный естественный язык применяется для понятного, достаточно строгого описания спецификаций процессов. При его использовании приняты следующие соглашения:

- логика процесса выражается в виде комбинации последовательных конструкций, конструкций выбора и итераций;
- глаголы должны быть активными, недвусмысленными и ориентированными на целевое действие (заполнить, вычислить, извлечь, а не модернизировать, обработать);
- логика процесса должна быть выражена четко и недвусмысленно.

При построении иерархии DFD переходить к детализации процессов следует только после определения содержания всех потоков и накопителей данных, которое описывается при помощи структур данных. Для каждого потока данных формируется список всех его элементов данных, затем элементы данных объединяются в структуры данных, соответствующие более крупным объектам данных (например, строкам документов или объектам предметной области). Каждый объект должен состоять из элементов, являющихся его атрибутами. Структуры данных могут содержать альтернативы, условные вхождения и итерации. Условное вхождение означает, что данный компонент может отсутствовать

в структуре (например, структура «данные о страховании» для объекта «служащий»). Альтернатива означает, что в структуру может входить один из перечисленных элементов. Итерация означает вхождение любого числа элементов в указанный диапазон (например, элемент «имя ребенка» для объекта «служащий»). Для каждого элемента данных может указываться его тип (непрерывные или дискретные данные). Для непрерывных данных могут указываться единица измерения, диапазон значений, точность представления и форма физического кодирования. Для дискретных данных может указываться таблица допустимых значений.

После построения законченной модели системы ее необходимо верифицировать (проверить на полноту и согласованность). В полной модели все ее объекты (подсистемы, процессы, потоки данных) должны быть подробно описаны и детализированы. Выявленные недетализированные объекты следует детализировать, вернувшись на предыдущие шаги разработки. В согласованной модели для всех потоков данных и накопителей данных должно выполняться правило сохранения информации: все поступающие куда-либо данные должны быть считаны, а все считываемые данные должны быть записаны.

При моделировании бизнес-процессов диаграммы потоков данных (DFD) используются для построения моделей «AS-IS» и «AS-TO-BE», отражая, таким образом, существующую и предлагаемую структуру бизнес-процессов организации и взаимодействие между ними. При этом описание используемых в организации данных на концептуальном уровне, независимо от средств реализации базы данных, выполняется с помощью модели «сущность-связь».

Ниже перечислены основные виды и последовательность работ при построении бизнес-моделей с использованием методики Йордона:

1. *Описание контекста процессов и построение начальной контекстной диаграммы.*

Начальная контекстная диаграмма потоков данных должна содержать нулевой процесс с именем, отражающим деятельность организации, внешние сущности, соединенные с нулевым процессом посредством потоков данных. Потоки данных соответствуют документам, запросам или сообщениям, которыми внешние сущности обмениваются с организацией.

2. *Спецификация структур данных.*

Определяется состав потоков данных и готовится исходная информация для построения концептуальной модели данных в виде струк-

тур данных. Выделяются все структуры и элементы данных типа «итерация», «условное вхождение» и «альтернатива». Простые структуры и элементы данных объединяются в более крупные структуры. В результате для каждого потока данных должна быть сформирована иерархическая (древовидная) структура, конечные элементы (листья) которой являются элементами данных, узлы дерева являются структурами данных, а верхний узел дерева соответствует потоку данных в целом.

3. *Построение начального варианта концептуальной модели данных.*

Для каждого класса объектов предметной области выделяется сущность. Устанавливаются связи между сущностями и определяются их характеристики. Строится диаграмма «сущность-связь» (без атрибутов сущностей).

4. *Построение диаграмм потоков данных нулевого и последующих уровней.*

Для завершения анализа функционального аспекта деятельности организации детализируется (декомпозируется) начальная контекстная диаграмма. При этом можно построить диаграмму для каждого события, поставив ему в соответствие процесс и описав входные и выходные потоки, накопители данных, внешние сущности и ссылки на другие процессы для описания связей между этим процессом и его окружением. После этого все построенные диаграммы сводятся в одну диаграмму нулевого уровня.

Процессы разделяются на группы, которые имеют много общего (работают с одинаковыми данными и/или имеют сходные функции). Они изображаются вместе на диаграмме более низкого (первого) уровня, а на диаграмме нулевого уровня объединяются в один процесс. Выделяются накопители данных, используемые процессами из одной группы.

Декомпозируются сложные процессы и проверяется соответствие различных уровней модели процессов.

Накопители данных описываются посредством структур данных, а процессы нижнего уровня — посредством спецификаций.

5. *Уточнение концептуальной модели данных.*

Определяются атрибуты сущностей. Выделяются атрибуты-идентификаторы. Проверяются связи, выделяются (при необходимости) связи «супертип-подтип».

Проверяется соответствие между описанием структур данных и концептуальной моделью



(все элементы данных должны присутствовать на диаграмме в качестве атрибутов).

### 3.4. Метод ARIS

В настоящее время наблюдается тенденция интеграции разнообразных методов моделирования и анализа систем, проявляющаяся в форме создания интегрированных средств моделирования. Одним из таких средств является продукт, носящий название ARIS (Architecture of Integrated Information System), разработанный германской фирмой IDS Scheer [Каменнова-01, Репин-04].

Система ARIS представляет собой комплекс средств анализа и моделирования деятельности предприятия. Ее методическую основу составляет совокупность различных методов моделирования, отражающих разные взгляды на исследуемую систему. Одна и та же модель может разрабатываться с использованием нескольких методов, что позволяет использовать ARIS специалистам с различными теоретическими знаниями и настраивать его на работу с системами, имеющими свою специфику.

Методика моделирования ARIS основывается на разработанной профессором Августом Шером теории построения интегрированных ИС, определяющей принципы визуального отображения всех аспектов функционирования анализируемых компаний. ARIS поддерживает четыре типа моделей, отражающих различные аспекты исследуемой системы:

- *организационные модели*, представляющие структуру системы — иерархию организационных подразделений, должностей и конкретных лиц, связи между ними, а также территориальную привязку структурных подразделений;
- *функциональные модели*, содержащие иерархию целей, стоящих перед аппаратом управления, с совокупностью деревьев функций, необходимых для достижения поставленных целей;
- *информационные модели*, отражающие структуру информации, необходимой для реализации всей совокупности функций системы;
- *модели управления*, представляющие комплексный взгляд на реализацию бизнес-процессов в рамках системы.

Для построения перечисленных типов моделей используются как собственные методы моделирования ARIS, так и различные известные методы и языки моделирования, в частности, UML.

В процессе моделирования каждый аспект деятельности предприятия сначала рассматривается отдельно, а после детальной проработки всех аспектов строится интегрированная модель, отражающая все связи между различными аспектами.

ARIS не накладывает ограничений на последовательность построения указанных выше типов моделей. Процесс моделирования можно начинать с любого из них, в зависимости от конкретных условий и целей, преследуемых разработчиками.

Модели в ARIS представляют собой диаграммы, элементами которых являются разнообразные объекты — «функция», «событие», «структурное подразделение», «документ» и т.п. Между объектами устанавливаются разнообразные связи. Так, между объектами «функция» и «структурное подразделение» могут быть установлены связи следующих видов:

- выполняет;
- принимает решение;
- участвует в выполнении;
- должен быть проинформирован о результатах;
- консультирует исполнителей;
- принимает результаты.

Каждому объекту соответствует определенный набор атрибутов, которые позволяют ввести дополнительную информацию о конкретном объекте. Значения атрибутов могут использоваться при имитационном моделировании или для проведения стоимостного анализа.

Таким образом, по результатам выполнения этого этапа возникает набор взаимосвязанных моделей, представляющих собой исходный материал для дальнейшего анализа.

Основная бизнес-модель ARIS — eEPC (extended Event-driven Process Chain — расширенная модель цепочки процессов, управляемых событиями). В табл. 2 приводятся основные объекты, используемые в данной нотации.

Помимо указанных в таблице основных объектов, при построении диаграммы eEPC могут быть использованы многие другие объекты. По существу, модель eEPC расширяет возможности IDEF0, IDEF3 и DFD, обладая всеми их достоинствами и недостатками. Применение большого числа различных объектов, связанных различными типами связей, значительно увеличивает размер модели и делает ее плохо читаемой. Для понимания смысла нотации eEPC достаточно рассмотреть основные типы объектов и связей. На рис. 12 представлена простейшая модель

Наименование объекта	Описание
Функция	Служит для описания функций (процедур, работ), выполняемых подразделениями/сотрудниками предприятия.
Событие	Служит для описания реальных событий, воздействующих на выполнение функций.
Организационная единица	Представляет различные организационные звенья предприятия (например, управление или отдел).
Документ	Отражает реальные носители информации, например, бумажный документ.
Прикладная система	Отражает реальную прикладную систему, поддерживающую выполнение функции.
Кластер информации	Характеризует данные (набор сущностей и связей между ними). Используется для создания моделей данных.
Связь между объектами	Описывает тип отношений между некоторыми объектами, например, активацию выполнения функции некоторым событием.
Логический оператор	Оператор одного из трех типов («И», «ИЛИ», исключающее «ИЛИ»), определяющий связи между событиями и функциями в рамках процесса. Позволяет описать ветвление процесса.

Табл. 2 Объекты модели eEPC

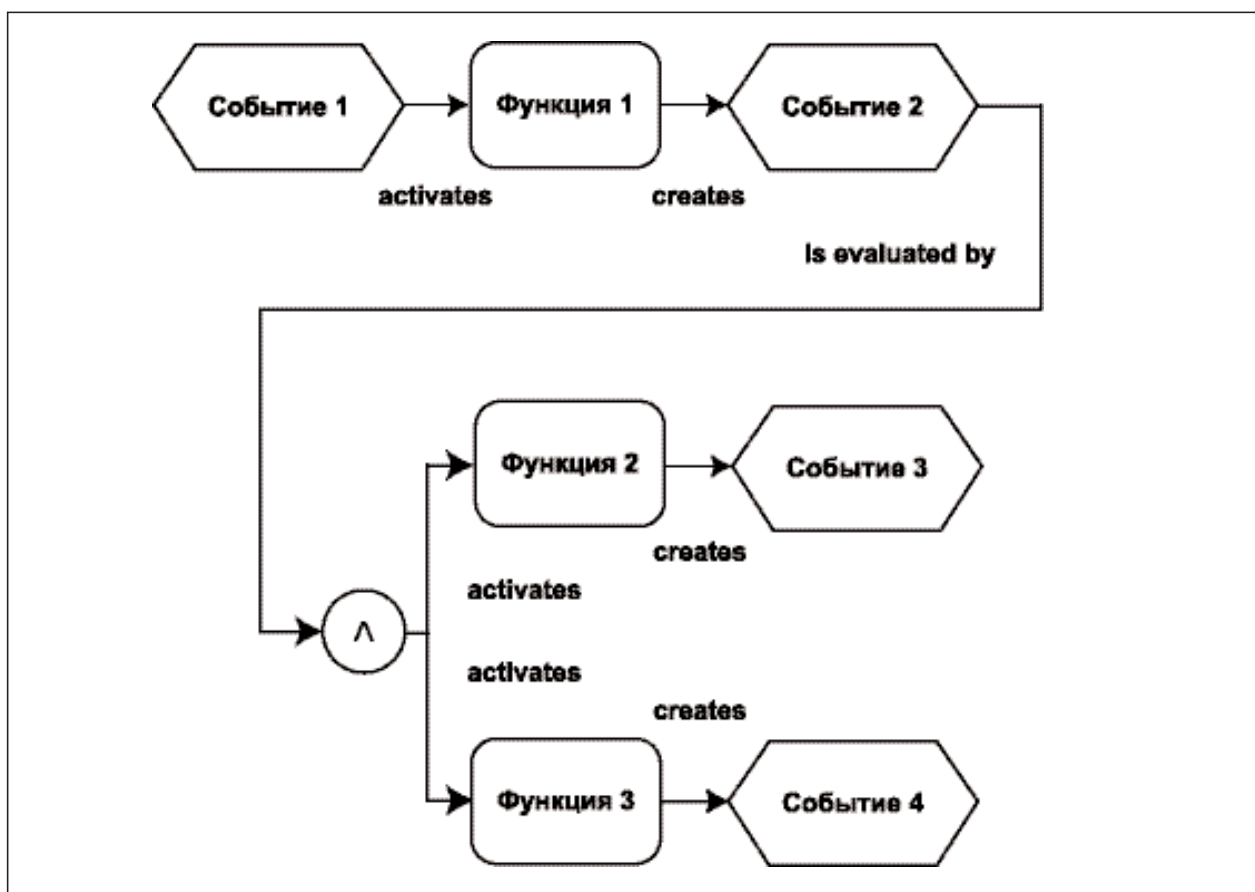


Рис. 12 Модель eEPC

eEPC, описывающая фрагмент бизнес-процесса предприятия.

На рис. 12 видно, что связи между объектами имеют определенный смысл и отражают последовательность выполнения функций в рамках процесса. Стрелка, соединяющая Событие 1 и Функцию 1, «активирует» или инициирует

выполнение Функции 1. Функция 1 «создает» Событие 2, за которым следует символ логического «И», «запускающий» выполнение Функций 2 и 3. Нотация eEPC построена на определенных правилах:

- каждая функция должна быть инициирована событием и должна завершаться событием;

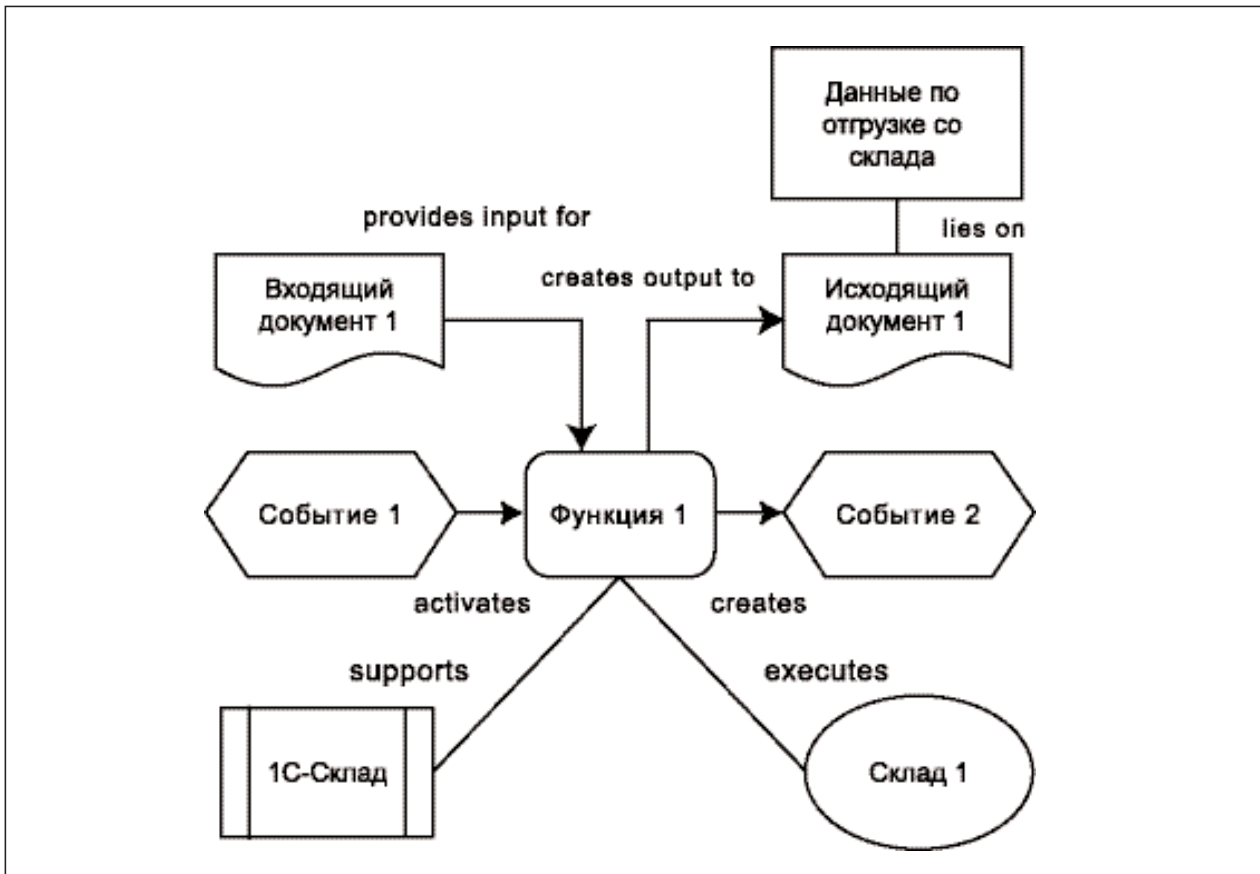


Рис. 13 Фрагмент модели бизнес-процесса

- в каждую функцию не может входить более одной стрелки, «запускающей» выполнение функции, и выходить не более одной стрелки, описывающей завершение выполнения функции.

На рис. 13 показано применение различных объектов ARIS при создании модели бизнес-процесса.

Из рис. 12 и 13 видно, что бизнес-процесс в нотации eEPC представляет собой поток последовательно выполняемых работ (процедур, функций), расположенных в порядке их выполнения. Реальная длительность выполнения процедур в eEPC визуально не отражается. Это приводит к тому, что при создании моделей возможны ситуации, когда на одного исполнителя будет возложено выполнение двух задач одновременно. Используемые при построении модели символы логики позволяют отразить ветвление и слияние бизнес-процесса. Для получения информации о реальной длительности процессов необходимо использовать другие инструменты описания, например, графики Ганта в системе MS Project.

Основное достоинство метода ARIS заключается в его комплексности, которая прояв-

ляется во взаимосвязи между моделями различных типов. Метод ARIS позволяет описывать деятельность организации с разных точек зрения и устанавливать связи между различными моделями. Однако такой подход трудно реализуем на практике, поскольку влечет за собой большой расход ресурсов (человеческих и финансовых) в течение длительного времени. Кроме того, инструментальная среда ARIS достаточно дорогостояща и сложна в использовании.

### 3.5. Метод Ericsson-Penker и образцы моделирования бизнес-процессов

Метод Ericsson-Penker [Eriksson-2000] представляет интерес прежде всего в связи с попыткой применения языка объектного моделирования UML [Буч-2000] (изначально предназначенного для моделирования архитектуры систем ПО) для моделирования бизнес-процессов. Это стало возможным благодаря наличию в UML механизмов расширения.

Механизмы расширения UML предназначены для того, чтобы разработчики могли адаптировать язык моделирования к своим конкретным нуждам, не меняя при этом его метамодель.

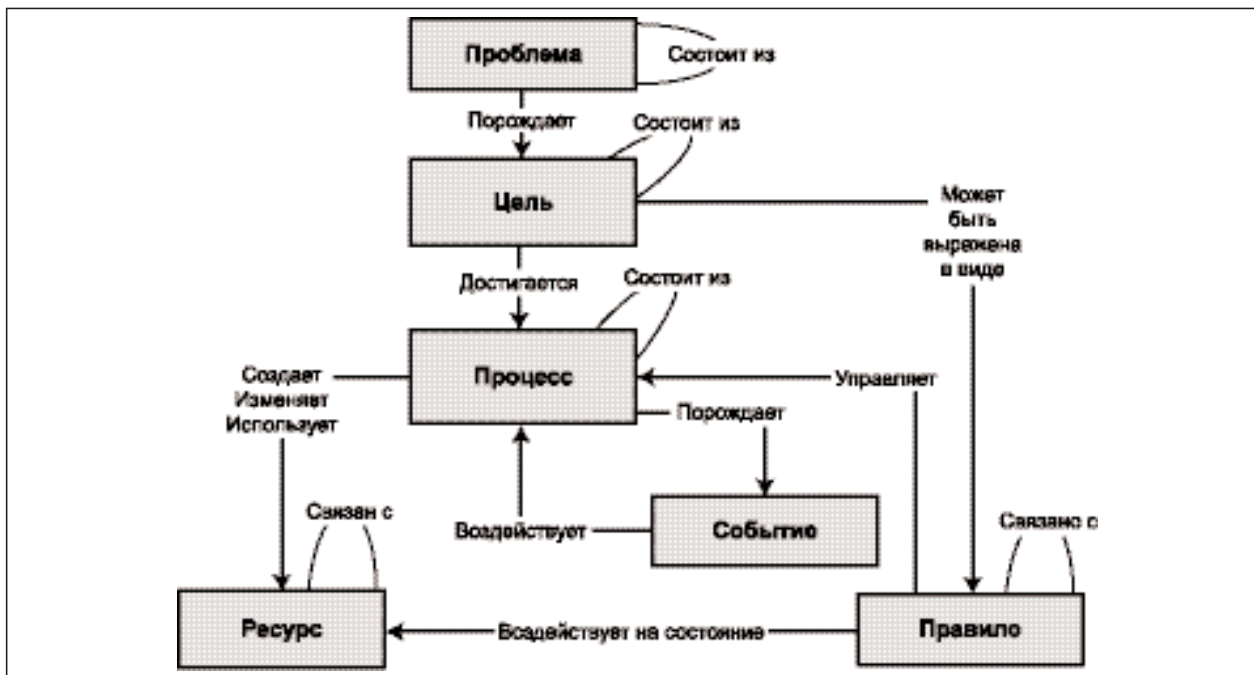


Рис. 14 Метамодел категории бизнес-модели

Наличие механизмов расширения принципиально отличает UML от таких средств моделирования, как IDEF0, IDEF1X, IDEF3, DFD и др. Перечисленные языки моделирования можно определить как сильно типизированные (по аналогии с языками программирования), поскольку они не допускают произвольной интерпретации семантики элементов моделей. UML, допуская такую интерпретацию (в основном за счет стереотипов), является слабо типизированным языком. К его механизмам расширения относятся:

- стереотипы;
- тегированные (именованные) значения;
- ограничения.

**Стереотип** — это новый тип элемента модели, который определяется на основе уже существующего элемента. Стереотипы расширяют нотацию модели, могут применяться к любым элементам модели и представляются в виде текстовой метки или пиктограммы. Стереотипы классов — это механизм, позволяющий разделять классы на категории. Участники проекта (аналитики) могут создавать свои собственные наборы стереотипов, формируя тем самым специализированные подмножества UML (например, для описания бизнес-процессов, Web-приложений, баз данных и т.д.). Такие подмножества (наборы стереотипов) в стандарте языка UML носят название **профилей** языка.

**Именованное значение** — это пара строк «тег = значение» или «имя = содержимое», в

которых хранится дополнительная информация о каком-либо элементе системы, например, время создания, статус разработки или тестирования, время окончания работы над ним и т.п.

**Ограничение** — это семантическое ограничение, имеющее вид текстового выражения на естественном или формальном языке (OCL — Object Constraint Language), которое невозможно выразить с помощью графической нотации UML.

Авторы метода Ericsson-Penker создали свой профиль UML для моделирования бизнес-процессов под названием Ericsson-Penker Business Extensions, введя набор стереотипов, описывающих процессы, ресурсы, правила и цели деятельности организации.

Метод использует четыре основные категории бизнес-модели:

- **Ресурсы** — различные объекты, используемые или участвующие в бизнес-процессах (люди, материалы, информация или продукты). Ресурсы структурированы, взаимосвязаны и подразделяются на физические, абстрактные, информационные и человеческие.
- **Процессы** — виды деятельности, изменяющие состояние ресурсов в соответствии с бизнес-правилами.
- **Цели** — назначение бизнес-процессов. Цели могут быть разбиты на подцели и соотносены с отдельными процессами. Цели достигаются в процессах и выражают требуемое состояние ресурсов. Цели могут быть выражены в виде одного или более правил.



- **Бизнес-правила** — условия или ограничения выполнения процессов (функциональные, поведенческие или структурные). Правила могут диктоваться внешней средой (инструкциями или законами) или могут быть определены в пределах бизнес-процессов. Правила могут быть определены с использованием языка OCL, который является частью стандарта UML.

Все эти категории связаны между собой: правило может определять способ структурирования ресурсов, ресурс назначается конкретному процессу, цель связана с выполнением конкретного процесса. Метамодел, определяющая связи между категориями, приведена на рис. 14.

Основной диаграммой UML, используемой в данном методе, является диаграмма деятельности (рис. 15).

Основным элементом диаграммы является **деятельность** (activity). Интерпретация этого термина зависит от той точки зрения, с которой строится диаграмма (это может быть некоторая задача, которую необходимо выполнить вручную или автоматизированным способом, или операция класса). Деятельность изображается в виде закругленного прямоугольника с текстовым описанием.

Любая диаграмма деятельности должна иметь начальную точку, определяющую начало потока событий. Конечная точка необязательна. На диаграмме может быть несколько конечных точек, но только одна начальная.

На диаграмме могут присутствовать **объекты** и **потоки объектов** (object flow). Объект может использоваться или изменяться в одной из деятельностей. Показ объектов и их состояний (в дополнение к диаграммам состояний UML) помогает понять, когда и как происходит смена состояний объекта.

Объекты связаны с деятельностями через потоки объектов. Поток объектов отмечается пунктирной стрелкой от деятельности к изменяемому объекту или от объекта к деятельности, использующей объект.

В примере на рис. 15 после ввода пользователем информации о кредитной карточке билет переходит в состояние «не подтвержден».

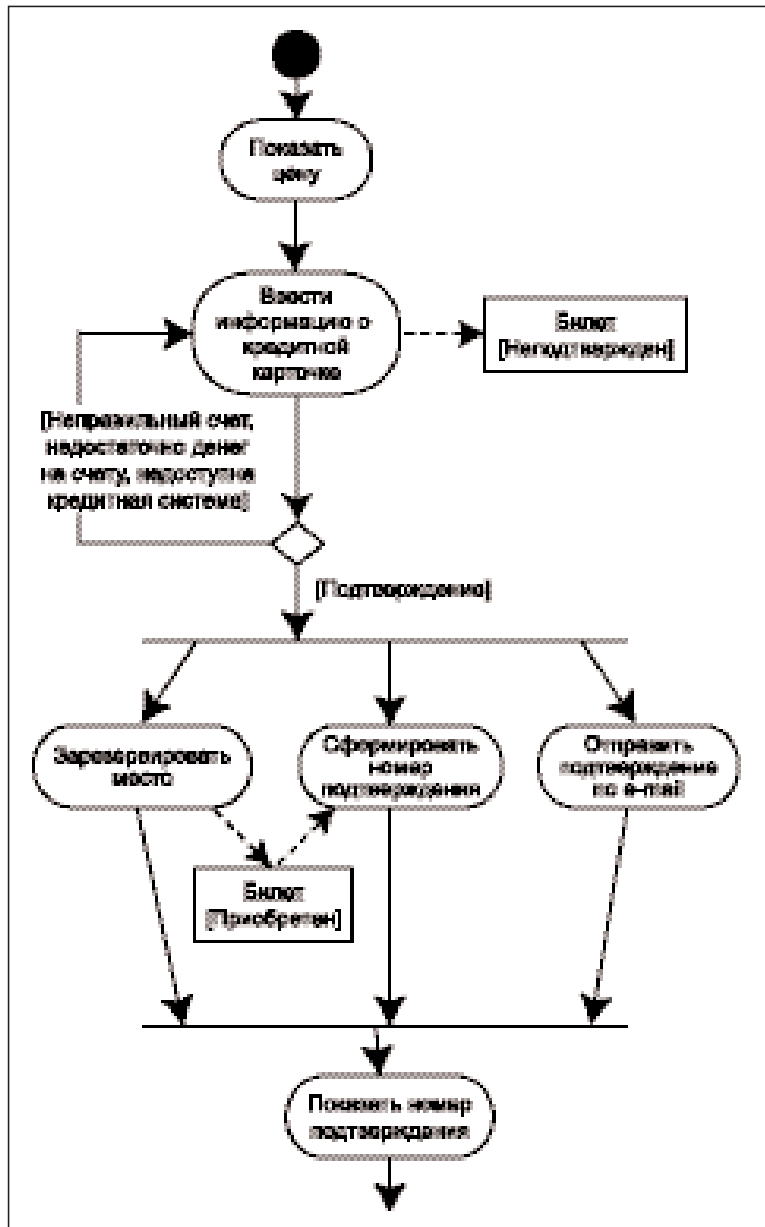


Рис. 15 Диаграмма деятельности

Когда завершится процесс обработки кредитной карточки и будет подтвержден перевод денег, возникает деятельность «зарезервировать место», переводящая билет в состояние «приобретен», и затем он используется в деятельности «формирование номера подтверждения».

**Переход** (стрелка) показывает, как поток управления переходит от одной деятельности к другой. Если для перехода определено событие, то переход выполняется только после наступления такого события. Ограничивающие условия определяют, когда переход может, а когда не может осуществиться.

Если необходимо показать, что две или более ветвей потока выполняются параллельно, ис-

пользуются **линейки синхронизации**. В данном примере параллельно выполняются резервирование места, формирование номера подтверждения и отправка почтового сообщения, а после завершения всех трех процессов пользователю выводится номер подтверждения.

Любая деятельность может быть подвергнута дальнейшей декомпозиции. Описание декомпозированной деятельности может быть представлено в виде другой диаграммы деятельности.

Подобно большинству других средств, моделирующих поведение некоторых объектов, диаграммы деятельности отражают только вполне определенные его аспекты, поэтому их лучше всего использовать в сочетании с другими средствами.

Бизнес-процесс в самом простом виде может быть описан как множество деятельностей. Метод Eriksson-Penker представляет образец процесса на диаграмме деятельности (рис. 16) в виде деятельности со стереотипом «process» (в качестве основы данного образца использовано представление процесса в методе IDEF0, расширенное за счет введения цели процесса). Процесс использует входные ресурсы и формирует выходные ресурсы, показанные в виде объектов со стереотипом «resource», соединенных с процессом связями зависимости. Ресурсы, играющие в методе IDEF0 роли «управления» и «механизма», также соединены с процессом связями зависимости со стереотипами «supply» и «control». Цель процесса показана как объект со стереотипом «goal».

Полная бизнес-модель включает множество представлений. Каждое представление выражено в одной или более диаграммах. Диаграммы могут иметь различные типы и изображать процессы, правила, цели и ресурсы во взаимодействиях друг с другом. Метод Eriksson-Penker использует четыре различных представления бизнес-модели:

- концептуальное представление — структура целей и проблем (дерево целей, представленное в виде диаграммы объектов);
- представление процессов — взаимодействие между процессами и ресурсами (в виде набора диаграмм деятельности);

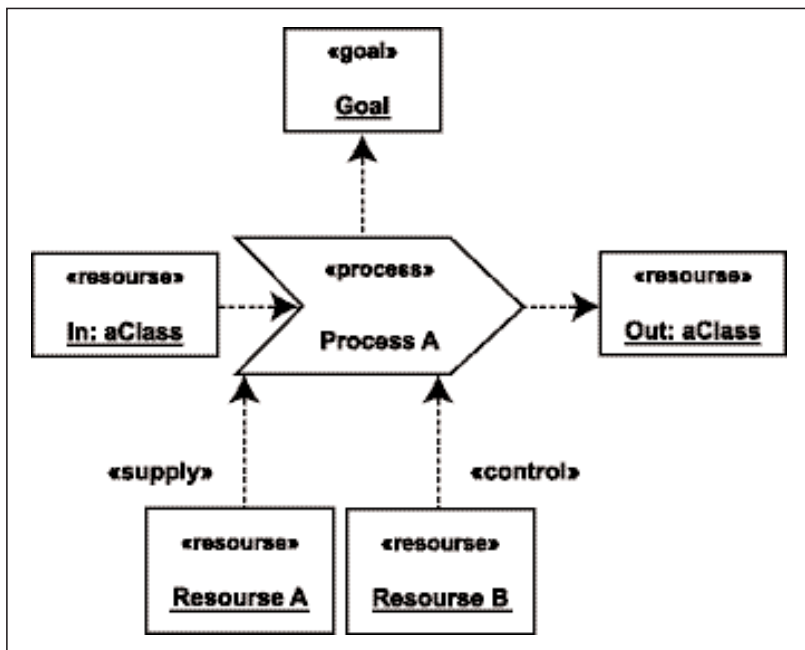


Рис. 16 Диаграмма деятельности для процесса

- структурное представление — структура организации и ресурсов (в виде диаграмм классов);
- представление поведения — поведение отдельных ресурсов и детализация процессов (в виде диаграмм деятельности, состояний и взаимодействия).

Метод Ericsson-Penker активно использует набор образцов моделирования бизнес-процессов. Образец (pattern) можно определить как общее решение некоторой проблемной ситуации в заданном контексте. Образец состоит из четырех основных элементов:

- имя;
- проблема;
- решение;
- следствия.

Сославшись на **имя образца**, можно сразу описать проблему, ее решения и их последствия. С помощью словаря образцов можно вести обсуждение с коллегами, упоминать образцы в документации, в тонкостях представлять проект системы.

**Проблема** — это описание решаемой задачи. Необходимо сформулировать задачу и ее контекст. Также может включаться перечень условий, при выполнении которых имеет смысл применять данный образец.

**Решение** — это описание элементов решения, связей между ними и функций каждого элемента. Конкретное решение или реализация

не имеются в виду, поскольку образец — это шаблон, применимый в самых разных ситуациях. Обычно дается абстрактное описание задачи и того, как она может быть решена с помощью некоего весьма обобщенного сочетания элементов (классов и объектов).

**Следствия** — это описание области применения, достоинств и недостатков образца. Хотя при описании решений о следствиях часто не упоминают, знать о них необходимо, чтобы можно было выбрать между различными вариантами и оценить преимущества и недостатки применения данного образца.

В языке UML образец представляется с помощью кооперации со стереотипом «pattern». **Кооперация** (collaboration) определяется как описание совокупности взаимодействующих объектов, реализующих некоторое поведение (например, в рамках варианта использования или операции класса). Кооперация имеет статическую и динамическую части. В статической части (на диаграмме классов) описываются роли, которые могут играть объекты и связи в экземпляре данной кооперации. Динамическая часть состоит из одной или более диаграмм взаимодействия, показывающих потоки сообщений, которыми обмениваются участники кооперации. Кроме того, любой образец содержит стандартную диаграмму классов под названием «Participants» («Участники»), на которой изображается сам образец в виде кооперации с его именем и набор классов, участвующих в реализации образца.

В качестве примера приведем образец бизнес-моделирования под названием Employment (Занятость).

**Проблема** заключается в моделировании различных форм занятости в пределах организации. Данная задача решается в контексте системы планирования ресурсов предприятия.

**Решение:** занятость моделируется как контракт между личностью и организацией, указывающий выполняемые обязанности, контрактные условия, даты начала и конца работы. Личность характеризуется набором атрибутов (имя, адрес и дата рождения), может занимать более чем одну должность в одной и той же организации.

На рис. 17 приведена диаграмма «Участники» для данного образца (примеры моделей здесь и далее приводятся в среде CASE-средства

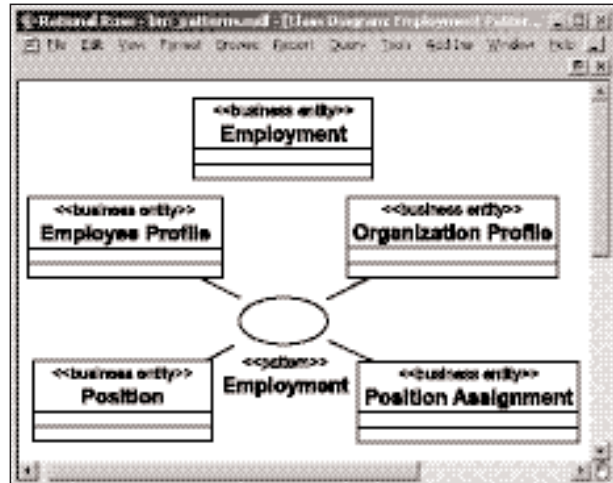


Рис. 17 Диаграмма «Участники» для образца Employment

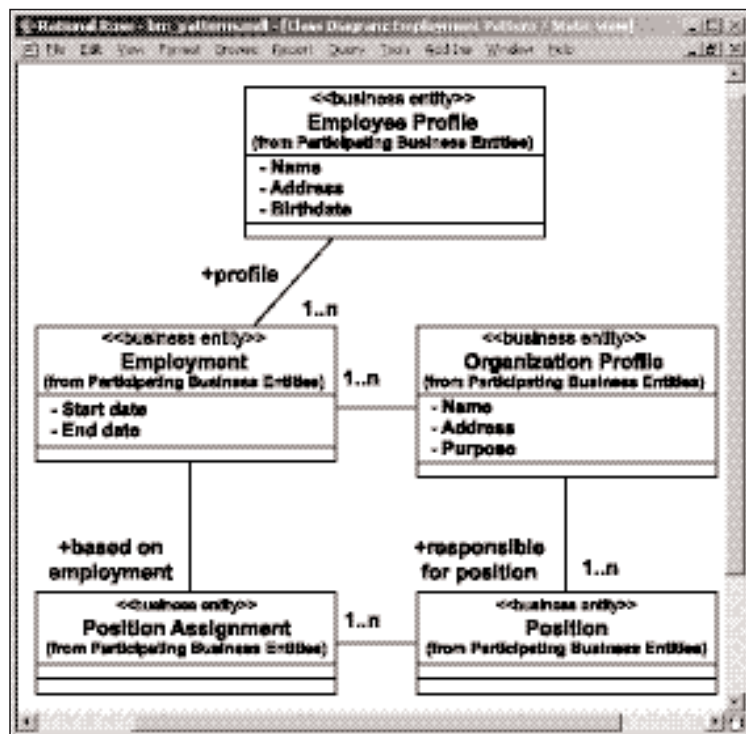


Рис. 18 Статическая часть образца Employment

Rational Rose). Она содержит кооперацию Employment и набор из пяти классов:

- Employee Profile (Служащий) — описание служащего с набором атрибутов.
- Organization Profile (Организация) — описание самой организации.
- Employment (Занятость) — описание связи между служащим и организацией.
- Position (Должность) — описание должности со своими атрибутами (такими, как должностной оклад и должностные инструкции).

- Position Assignment (Назначение на должность) — описание связи между служащим и занимаемыми должностями.

Статическая часть образца (диаграмма классов) показана на рис. 18.

### 3.6. Метод моделирования, используемый в технологии Rational Unified Process

Язык UML используется также в методе моделирования бизнес-процессов, являющемся частью технологии Rational Unified Process [Крачтен-02] компании IBM Rational Software. Этот метод, направленный прежде всего на создание основы для формирования требований к ПО, предусматривает построение двух базовых моделей:

- модели бизнес-процессов (Business Use Case Model);
- модели бизнес-анализа (Business Analysis Model).

**Модель бизнес-процессов** — модель, описывающая бизнес-процессы организации в терминах ролей и их потребностей. Она представляет собой расширение модели вариантов использования (use case) UML [Коберн-02] за счет введения набора стереотипов — Business Actor (стереотип действующего лица) и Business Use Case (стереотип варианта использования).

**Business Actor** (действующее лицо бизнес-процессов) — это некоторая роль, внешняя по отношению к бизнес-процессам организации. Потенциальными кандидатами в действующие лица бизнес-процессов являются:

- акционеры;
- заказчики;
- поставщики;
- партнеры;
- потенциальные клиенты;
- местные органы власти;
- сотрудники подразделений организации, деятельность которых не охвачена моделью;
- внешние системы.

Список действующих лиц составляется путем ответа на следующие вопросы:

- Кто извлекает пользу из существования организации?
- Кто помогает организации осуществлять свою деятельность?
- Кому организация передает информацию и от кого получает?

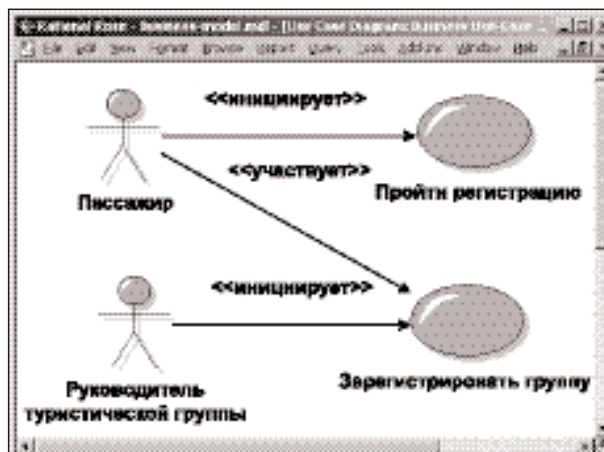


Рис. 19 Диаграмма вариантов использования для процесса регистрации пассажиров в аэропорту

**Business Use Case** (вариант использования с точки зрения бизнес-процессов) определяется как описание последовательности действий (потока событий) в рамках некоторого бизнес-процесса, приносящей ощутимый результат конкретному действующему лицу.

Это определение подобно общему определению бизнес-процесса, но имеет более точный смысл. В терминах объектной модели Business Use Case представляет собой класс, объектами которого являются конкретные потоки событий в рамках описываемого бизнес-процесса.

Данный метод концентрирует внимание в первую очередь на элементарных бизнес-процессах. Такой процесс можно определить как задачу, выполняемую одним человеком в одном месте в одно время в ответ на некоторое событие, приносящую конкретный результат и переводящую данные в некоторое устойчивое состояние (например, подтверждение платежа по кредитной карточке). Выполнение такой задачи обычно включает от пяти до десяти шагов и может занимать от нескольких минут до нескольких дней, но рассматривается как один сеанс взаимодействия действующего лица с исполнителями.

Каждый Business Use Case отражает цель или потребность некоторого действующего лица. Например, если рассмотреть процесс регистрации пассажиров в аэропорту (рис. 19), то его основным действующим лицом будет сам Пассажир, главная цель которого в данном процессе — пройти регистрацию. Эта цель моделируется в виде Business Use Case с наименованием «Пройти регистрацию». Другим действующим лицом является Руководитель туристической группы, регистрирующий группу пассажиров. Стереотипы связей явно показывают роль дей-



ствующих лиц по отношению к вариантам использования.

Описание Business Use Case представляет собой спецификацию (текстовый документ), которая, подобно обычному варианту использования, состоит из следующих пунктов [Коберн-02]:

- наименование;
- краткое описание;
- цели и результаты (с точки зрения действующего лица);
- описание сценариев (основного и альтернативных);
- специальные требования (ограничения по времени выполнения или другим ресурсам);
- расширения (исключительные ситуации);
- связи с другими Business Use Case;
- диаграммы деятельности (для наглядного описания сценариев – при необходимости).

**Пример спецификации Business Use Case:**

*Наименование* – пройти регистрацию.

*Краткое описание* – данный Business Use Case реализует процесс регистрации пассажира на рейс.

*Цели* – получить посадочный талон и сдать багаж.

*Основной сценарий:*

1. Пассажир встает в очередь к стойке регистратора.
2. Пассажир предъявляет билет регистратору.
3. Регистратор подтверждает правильность билета.
4. Регистратор оформляет багаж.
5. Регистратор резервирует место для пассажира.
6. Регистратор печатает посадочный талон.

7. Регистратор выдает пассажиру посадочный талон и квитанцию на багаж.
8. Пассажир уходит от стойки регистратора.

*Альтернативные сценарии:*

- 3а. Билет неправильно оформлен – регистратор отправляет пассажира к агенту по перевозкам.
- 4а. Багаж превышает установленный вес – регистратор оформляет доплату.

*Специальные требования* – время регистрации не должно превышать одной минуты.

Описание Business Use Case может сопровождаться целью процесса, которая так же, как и в методе Eriksson-Penker, моделируется с помощью класса со стереотипом «goal», а дерево целей изображается в виде диаграммы классов.

Для каждого Business Use Case строится **модель бизнес-анализа** – объектная модель, описывающая реализацию бизнес-процесса в терминах взаимодействующих объектов (бизнес-объектов – Business Object), принадлежащих к двум классам – Business Worker и Business Entity.

**Business Worker** (исполнитель) – активный класс, представляющий собой абстракцию исполнителя, выполняющего некоторые действия в рамках бизнес-процесса. Исполнители взаимодействуют между собой и манипулируют различными сущностями, участвуя в реализациях сценариев Business Use Case. На диаграмме классов UML исполнитель представляется в виде класса со стереотипом «business worker». Например, если рассмотреть Business Use Case «Пройти регистрацию», можно определить в нем двух исполнителей – Регистратора и Координатора багажа.

**Business Entity** (сущность) – пассивный класс, не инициирующий никаких взаимодействий. Объект такого класса может участвовать в реализациях различных Business Use Case. Сущность является объектом различных действий со стороны исполнителей.

Понятие Business Entity аналогично понятию сущности в модели «сущность-связь», за исключением того, что в данной модели не определяется поведение сущности, а в объектной модели сущность может иметь набор обязанностей. На диаграмме классов UML сущность представляется в виде класса со стереотипом «business entity». Например, в Business Use Case «Пройти ре-

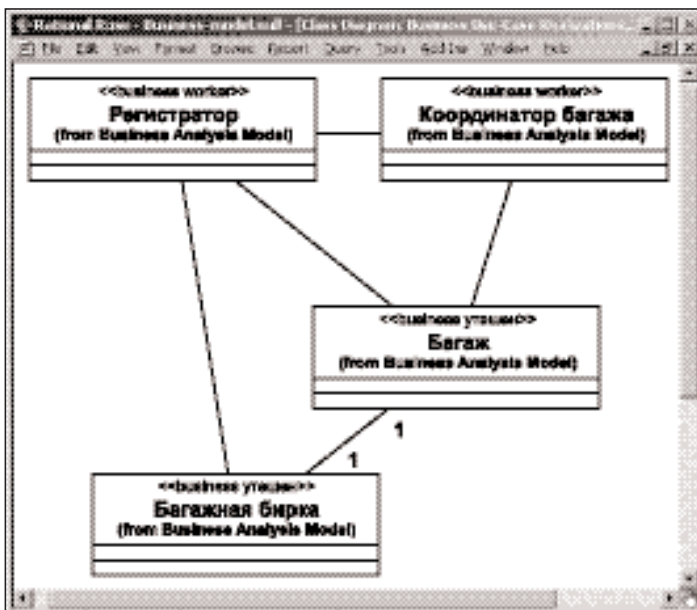


Рис. 20 Диаграмма классов модели бизнес-анализа



гистрацию» можно определить следующие сущности: Билет, Рейс, Авиакомпания, Багаж, Багажная бирка.

Модель бизнес-анализа может состоять из диаграмм разных типов. В состав модели обязательно должна входить диаграмма классов, содержащая исполнителей и сущности. Пример такой диаграммы для Business Use Case «Пройти регистрацию» приведен на рис. 20.

На данной диаграмме ассоциации между классами-исполнителями отражают наличие взаимодействия между реальными исполнителями (Регистратором и Координатором багажа). Ассоциации между классами-исполнителями и классами-сущностями показывают, какими именно объектами манипулируют конкретные исполнители (Регистратор имеет дело с Багажом и Багажной биркой, а Координатор багажа — только с Багажом). Ассоциации между классами-сущностями отражают различные структурные связи (к одному месту багажа прикрепляется одна багажная бирка).

Кроме диаграммы классов, модель бизнес-анализа может включать:

- Диаграммы последовательности (и кооперативные диаграммы), описывающие сценарии Business Use Case в виде последовательности обмена сообщениями между объектами-действующими лицами и объектами-исполнителями. Такие диаграммы помогают явно определить в модели обязанности каждого исполнителя в виде набора операций класса. Пример диаграммы последовательности, описывающей основной сценарий Business Use Case «Пройти регистрацию», приведен на рис. 21. Модифицированная диаграмма классов модели бизнес-анализа с операциями приведена на рис. 22.
- Диаграммы деятельности с потоками объектов и «плавательными дорожками», описывающие взаимосвязи между сценариями одного или различных Business Use Case. Пример

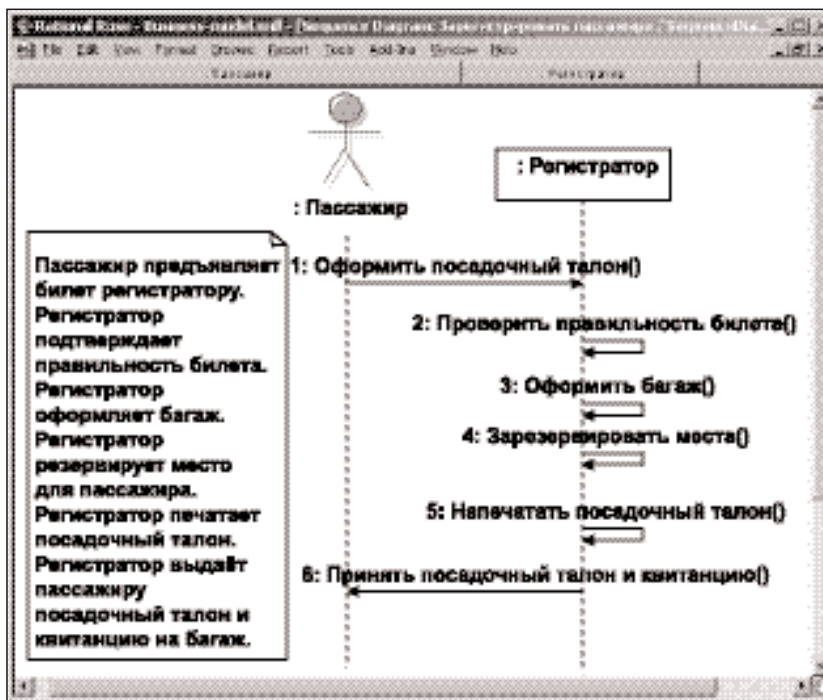


Рис. 21 Диаграмма последовательности для основного сценария Business Use Case «Пройти регистрацию»

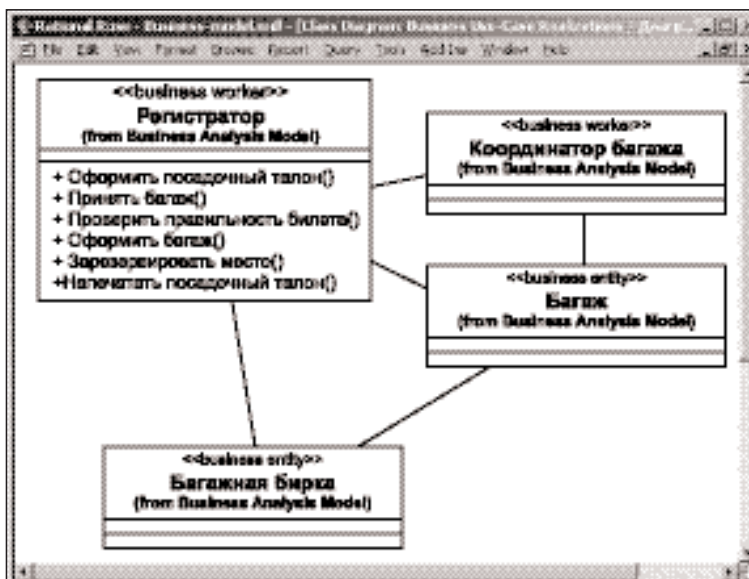


Рис. 22 Модифицированная диаграмма классов модели бизнес-анализа с операциями

такой диаграммы для процесса заказа товаров в торговой компании приведен на рис. 23.

- Диаграммы состояний, описывающие поведение отдельных бизнес-объектов (например, для сущности «Багаж» можно описать переходы между состояниями «Определен вес», «Зарегистрирован», «Находится на транспорте» и т.д.).

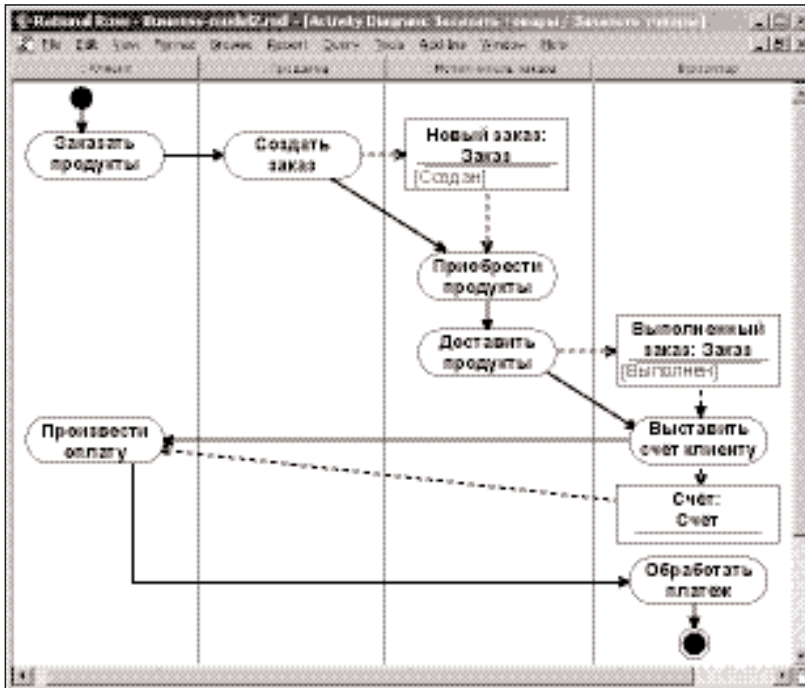


Рис. 23 Диаграмма деятельности для процесса заказа товаров

Метод моделирования Rational Unified Process предусматривает специальное соглашение, связанное с группировкой структурных элементов и диаграмм бизнес-модели. Это соглашение включает следующие правила:

- Все действующие лица, варианты использования и диаграммы вариантов использования для бизнес-процессов помещаются в пакет с именем Business Use Case Model.
- Все классы и диаграммы моделей бизнес-анализа помещаются в пакет с именем Business Analysis Model.
- Если моделируется деятельность более чем одного подразделения организации, то совокупность всех классов-исполнителей и классов-сущностей из моделей бизнес-анализа для различных Business Use Case разделяется на пакеты, соответствующие этим подразделениям. Этим пакетам присваиваются наименования подразделений (например, Бухгалтерия, Отдел доставки и т.п.) и стереотип «business system».
- Диаграммы модели бизнес-анализа, относящиеся к конкретному Business Use Case (диаграммы классов, последовательности, деятельности и состояний) помещаются в кооперацию с именем данного Business Use Case и стереотипом «business use-case realization» (реализация бизнес-процесса). Все коопера-

ции помещаются в пакет с именем Business Use Case Realizations.

Пример структуры бизнес-модели для процесса регистрации пассажиров в аэропорту приведен на рис. 24.

Метод моделирования Rational Unified Process обладает следующими достоинствами:

- модель бизнес-процессов строится вокруг участников процессов (заинтересованных лиц) и их целей, помогая выявить все потребности клиентов организации. Нетрудно заметить, что такой подход в наибольшей степени применим для организаций, работающих в сфере оказания услуг (торговые организации, банки, страховые компании и т.д.);
- моделирование на основе вариантов использования способствует хорошему пониманию бизнес-модели со стороны заказчиков;

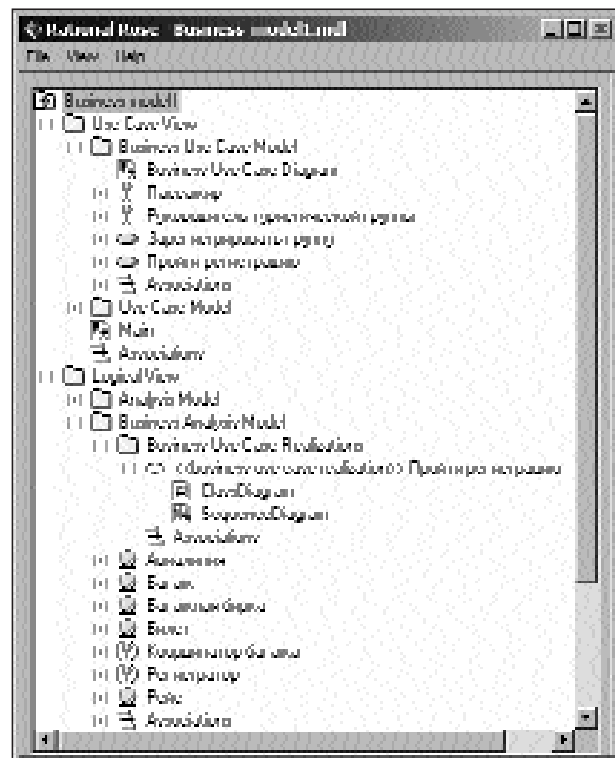


Рис. 24 Пример структуры бизнес-модели

- метод предусматривает достаточно простой переход от бизнес-модели к системным требованиям.

Однако следует отметить, что при моделировании деятельности крупной организации, занимающейся как производством продукции, так и оказанием услуг, необходимо применять различные методы моделирования, поскольку для моделирования производственных процессов более предпочтительным является процессный подход (например, метод Eriksson-Penker).

## 4. Сравнительный анализ различных методов и инструментальных средств моделирования

Основная задача сравнительного анализа состоит в том, чтобы ответить на ряд вопросов, возникающих у руководителей и специалистов в начале проекта по моделированию и реорганизации бизнес-процессов предприятия:

- Какое инструментальное средство использовать в проекте (ARIS, BPwin, Rational Rose и др.)?
- Какой метод использовать для описания процессов?
- Как моделировать процессы с использованием некоторого инструментального средства?

В настоящее время на российском рынке представлено достаточно большое количество инструментальных средств (ARIS, AllFusion Modeling Suite, Rational Rose и др.), которые позволяют, так или иначе, создавать описания (модели) бизнес-процессов. Рациональный выбор средств возможен при понимании руководством компании и ее специалистами нескольких аспектов:

- целей проекта;
- требований к информации о бизнес-процессах, необходимой для анализа и принятия решений в рамках конкретного проекта;
- возможностей инструментальных средств в части описания процессов.

Говорить о преимуществе того или иного метода и средств бессмысленно, пока не определены тип и рамки проекта, его основные задачи. Описание бизнес-процессов проводится с целью их дальнейшего анализа и реорганизации. Целью реорганизации может быть внедрение информационной системы, сокращение затрат на выпуск продукции, повышение качества обслуживания клиентов, создание должностных и рабочих инструкций при внедрении стандартов ISO-9000 и т.д. Для каждой такой задачи существуют определенные параметры, определяющие набор критических знаний по бизнес-процессу. От задачи к задаче требования к описанию бизнес-процессов могут меняться.

В качестве примера можно привести результаты сравнительного анализа методов ARIS и IDEF (IDEF0, IDEF3), а также поддерживающих их инструментальных средств ARIS Toolset и BPwin [Репин-04]. Итак...

Одним из важнейших аспектов описания моделей бизнес-процессов является отражение управляющих воздействий, обратных связей по контролю и управлению процедурой. В нотации ARIS eEPC управление процедурой может быть отражено только при помощи указания входящих документов, которые регламентируют выполнение процедуры, и последовательности выполнения процедур во времени (запускающие события). В отличие от ARIS, в нотации IDEF0 каждая процедура должна иметь хотя бы одно управляющее воздействие (вход управления — стрелка сверху). Если при создании модели в eEPC указывать только последовательность выполнения процедур, не заботясь об отражении управляющих документов и информации, полученные модели будут иметь низкую ценность с точки зрения анализа и дальнейшего использования. К сожалению, именно эта ошибка наиболее распространена на практике. Создается модель потока работ (workflow), отражающая простую последовательность выполнения процедур и входящих/исходящих документов, при этом управляющие (контрольные) воздействия на функции в модели не отражаются.

Кроме того, если попытаться в нотации ARIS eEPC отразить все условия и ограничения, определяющие выполнение функций, то потребуются описать большое количество событий и входящей информации (например, устных распоряжений руководителей), и модель станет сложной и плохо читаемой (эти недостатки присущи так же и нотации IDEF3). Указанных недостатков нет у нотации IDEF0. В то же время в

IDEF0 не предусмотрено использование символов логики выполнения процесса.

Таким образом, нотация ARIS eEPC является расширением достаточно простой нотации IDEF3. Для адекватного описания процесса управления в нотации eEPC необходимо заранее договориться, как будут отражены в модели документы (информация), регламентирующие выполнение процедур процесса.

Функциональные возможности инструментальных средств моделирования ARIS Toolset и BPwin можно корректно сравнивать только по отношению к определенному кругу задач. С точки зрения формирования моделей бизнес-процессов каждая из рассматриваемых систем имеет свои преимущества и недостатки. В зависимости от решаемых задач эти преимущества и недостатки могут как усиливаться, так и наоборот. Например, отсутствие четких соглашений по моделированию управляющих воздействий в рамках eEPC ARIS может привести к созданию моделей, не отвечающих на поставленные вопросы, в то время как нотация IDEF0 системы BPwin позволяет решить эту задачу. С другой стороны, описание процедуры, выполняемой одним сотрудником, может быть описано более адекватно при помощи eEPC ARIS, чем IDEF0 или IDEF3 BPwin.

Сравнивая две системы, следует отметить, что для хранения моделей в ARIS используется объектная СУБД, и под каждый проект создается новая база данных. Для удобства пользователя модели (объекты моделей) могут храниться в различных группах, организованных в зависимости от специфики проекта. Естественно, что в ARIS предусмотрены различные функции по администрированию базы данных: управление доступом, консолидация и т.п. В BPwin данные модели хранятся в файле, что существенно упрощает работу по созданию модели, но с другой стороны ограничивает возможности по анализу объектов модели. Для управления групповой разработкой используется средство Model Mart, обеспечивающее многопользовательский доступ к моделям, созданным с помощью ERwin и BPwin.

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

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

ARIS предоставляет существенно больше возможностей по работе с отдельными объектами модели, но именно вследствие чрезмерного количества настроек работа по созданию модели должна регламентироваться жесткими и объемными соглашениями по моделированию (стандартами). Разработка таких соглашений требует значительного времени (1-3 месяца) и высококвалифицированных специалистов. Если проект с использованием ARIS начинается без детальной проработки таких соглашений, то вероятность создания моделей бизнес-процессов, не отвечающих на поставленные вопросы, составляет 80-90%. В свою очередь, BPwin отличается простотой в использовании и достаточной строгой регламентацией при создании диаграмм (стандарт IDEF и рекомендации по его применению, бланк IDEF для создания диаграммы, ограниченное количество обязательно заполняемых полей, ограничение количества объектов на одной диаграмме и т.д.). ARIS, безусловно, является более мощным и «тяжелым» инструментом по сравнению с BPwin, но это в итоге оборачивается значительными трудностями и высокими затратами на его эксплуатацию.

Таким образом, для ведения небольших по масштабам (малые и средние предприятия, 2-5 человека в группе консультантов) и длительности (2-3 месяца) проектов рационально использовать BPwin. Для крупных и/или длительных проектов (например, внедрение системы непрерывного улучшения бизнес-процессов в соответствии со стандартами ISO) больше подходит ARIS. В этом случае подготовительные работы по созданию регламентирующей документации могут занять 1-3 месяца, но это является необходимым элементом последующей успешной работы.

По мнению автора, современные методы и инструментальные средства моделирования



достигли такого уровня, что их возможности с точки зрения изобразительных средств моделирования в настоящее время стали примерно одинаковыми. При этом одним из основных критериев выбора того или иного метода и инструмента становится степень владения им со стороны консультанта или аналитика, грамотность выражения своих мыслей на языке моделирования, обеспечивающая достаточный уровень понимания моделей со стороны руководителей и специалистов организации. В противном случае в моделях, построенных с использованием *любого* метода, будет невозможно разобраться.

## 5. Перспективные направления в моделировании бизнес-процессов

Как было сказано выше, в настоящее время предпринимаются многочисленные проекты, целью которых является интеграция существующих методов и языков моделирования и создание единого методического и технологического базиса моделирования бизнес-процессов, а в более широком контексте — моделирования предприятий (enterprise modeling) [BPMN-03, UEMML-02, BPD-03].

### 5.1. Деятельность консорциума Business Process Management Initiative (BPMI)

Консорциум BPMI был создан в августе 2000 г. по инициативе компании Intalio группой из шестнадцати компаний-разработчиков ПО и консалтинговых фирм. BPMI (<http://www.bpmi.org>) — независимая организация, занимающаяся разработкой открытых спецификаций для управления процессами электронной коммерции. К таким спецификациям относятся проекты стандартов Business Process Modeling Language (BPML) и Business Process Query Language (BPQL), предназначенных для управления бизнес-процессами (аналогично использованию SQL для управления данными с помощью СУБД). BPML — это метаязык для моделирования бизнес-процессов, также как XML — мета-

язык для моделирования данных. BPML позволяет создать абстрактную исполнимую модель взаимодействующих процессов, основанную на концепции конечного автомата.

В 2003 г. BPMI опубликовал проект стандарта Business Process Modeling Notation (BPMN) [BPMN-03]. Целью этого проекта является создание общей нотации для различных категорий специалистов: от бизнес-аналитиков и экспертов организаций до разработчиков ПО. BPMN состоит из одной диаграммы под названием Business Process Diagram (BPD) (рис. 25), которая непосредственно отображается в конструкции BPML.

Хотя спецификация BPMN в настоящее время существует только в версии 1.0, многие компании уже приняли ее на вооружение. BPMI не является комитетом по стандартизации, поэтому стандарт BPMN будет в конечном счете передан соответствующей организации. Наиболее вероятным кандидатом на роль такой организации является консорциум Object Management Group (OMG), и переговоры относительно такой передачи уже имели место. Учитывая высокую степень сходства между BPMN и диаграммой деятельности UML 2.0, можно допустить их интеграцию в будущем в общую модель.

### 5.2. Проект UEMML

Проект Unified Enterprise Modeling Language (UEML) [UEML-02], финансируемый Европейской Комиссией, был предпринят с целью интеграции многочисленных языков моделирования архитектуры предприятий (Enterprise Modeling Languages) и создания в перспективе унифицированного языка моделирования с четко определенными синтаксисом, семантикой и правилами отображений между различными средствами моделирования. Основой для такой интеграции послужили модели GERAM (Generalised Enterprise Reference Architecture and Methodology) и Захмана [UEML-02]. Проект UEMML включает разработку:

- общего визуального, основанного на шаблонах языка для коммерческих инструментальных средств моделирования;
- стандартных, независимых от инструментов механизмов передачи моделей между проектами;
- репозитория моделей предприятий.

Одним из результатов проекта, в частности, явилось создание портала <http://www.ueml.org>,



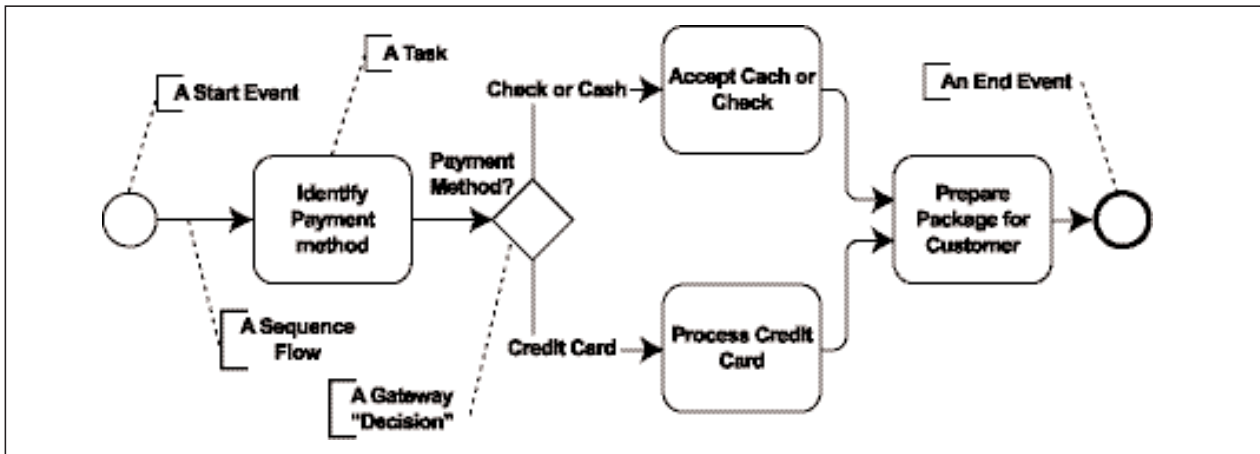


Рис. 25 Пример простейшей BPD

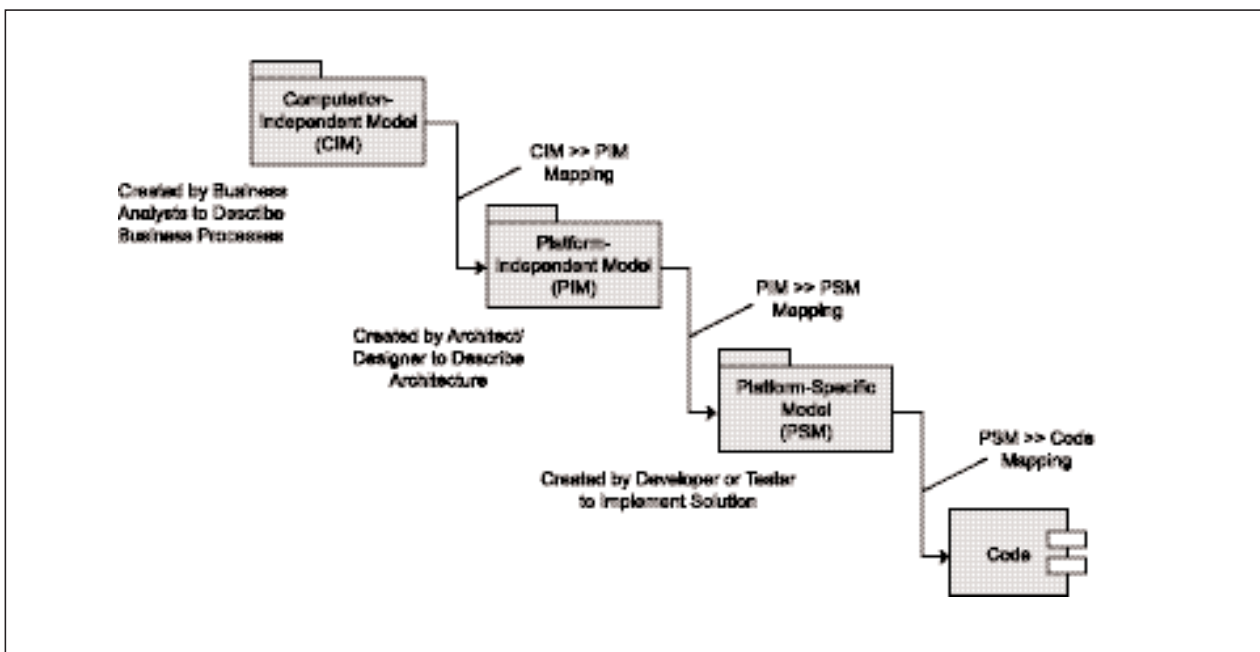


Рис. 26 Процесс создания моделей

который содержит всю информацию по данному проекту.

### 5.3. Работы в рамках проекта OMG MDA

OMG – это консорциум разработчиков ПО и пользователей, представляющих различные коммерческие, государственные и академические организации, насчитывающий около 800 участников. OMG занимается разработкой различных стандартов в области взаимодействия распределенных систем (наиболее известные из них – CORBA и UML).

Работа OMG в области моделирования бизнес-процессов связана в основном с концепцией Model Driven Architecture (MDA) [Кузнецов-03].

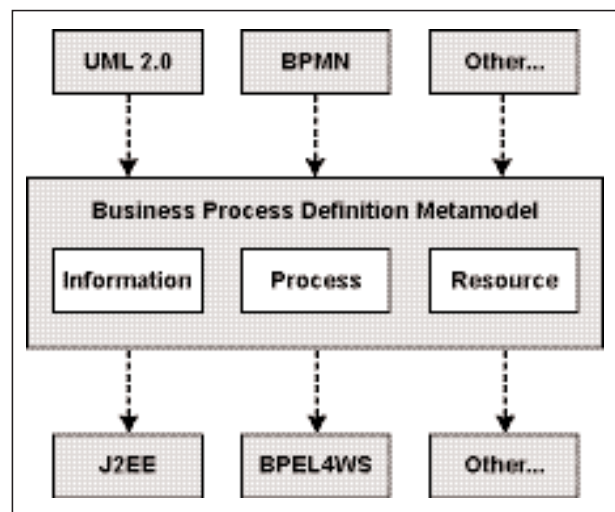


Рис. 27 Представление BPDM

MDA интегрирует различные подходы к моделированию и вводит набор отображений между моделями различных уровней абстракции (рис. 26). Любая организация, использующая MDA, может разрабатывать только те модели, которые требуются для ее собственных целей.

В настоящее время тремя главными инициативными проектами OMG являются создание метамodelей для описания бизнес-процессов (Business Process Definition Metamodel – BPDM) [BPDM-03], бизнес-правил (Business Semantics of Business Rules, and Production Rule Representation) и онтологии (Ontology Definition Metamodel). Назначение BPDM (рис. 27) – интеграция и обеспечение взаимодействия между моделями, используемыми различными организациями (такими, как диаграммы UML или BPMN). Предполагается, что BPDM будет реализована в виде профиля UML 2.0.

Аналогично, OMG работает над стандартизацией бизнес-правил и их совместимостью с BPDM. Все это вместе взятое должно в перспективе обеспечить новый уровень совместимости между моделями, используемыми для описания бизнес-процессов и ПО.

## Библиография

- [Буч-2000] Буч Г., Рамбо Дж., Джекобсон А. Язык UML. Руководство пользователя.: Пер. с англ. – М.: ДМК, 2000.
- [Калашян-03] Калашян А.Н., Калянов Г.Н. Структурные модели бизнеса: DFD-технологии. – М.: Финансы и статистика, 2003.
- [Каменнова-01] Каменнова М., Громов А., Феропонтов М., Шматалюк А. Моделирование бизнеса. Методология ARIS. – М.: Весть-Мета-Технология, 2001.
- [Коберн-02] Коберн А. Современные методы описания функциональных требований к системам.: Пер. с англ. – М.: ЛОРИ, 2002.
- [Крачтен-02] Крачтен Ф. Введение в Rational Unified Process.: Пер. с англ. – М.: Вильямс, 2002.
- [Кузнецов-03] Кузнецов М. MDA – новая концепция интеграции приложений. – «Открытые системы», №9, 2003.
- [Марка-93] Марка Д.А., МакГоуэн К. Методология структурного анализа и проектирования. – М.: МетаТехнология, 1993.
- [Ойхман-97] Ойхман Е.Г., Попов Э.В. Реинжиниринг бизнеса: реинжиниринг организации и информационные технологии. – М.: Финансы и статистика, 1997.
- [РД-2000] Методология функционального моделирования IDEF0. Руководящий документ РД IDEF0 – 2000. – М.: Госстандарт России, 2000.
- [Репин-04] Репин В.В., Елиферов В.Г. Процессный подход к управлению. Моделирование бизнес-процессов. – М.: РИА «Стандарты и качество», 2004.
- [Черемных-01] Черемных С.В., Семенов И.О., Ручкин В.С. Структурный анализ систем: IDEF-технологии. – М.: Финансы и статистика, 2001.
- [BPDM-03] Business Process Definition Metamodel. Request For Proposal. OMG Document: bei/2003-01. <http://www.omg.org>
- [BPMN-03] Business Process Modeling Notation. Working Draft (1.0) August 25, 2003. <http://www.bpmn.org>
- [Eriksson-2000] Eriksson, Hans-Erik and Penker, Magnus. Business Modeling with UML: Business Patterns at work. Wiley Computer Publishing, 2000.
- [UEML-02] Report on the State of the Art in Enterprise Modeling. Project UEML: Unified Enterprise Modeling Language. September 27th 2002. <http://www.ueml.org>