

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
УНИВЕРСИТЕТ ИТМО**

О.А. Цуканова

**МЕТОДОЛОГИЯ И ИНСТРУМЕНТАРИЙ МОДЕ-
ЛИРОВАНИЯ БИЗНЕС-ПРОЦЕССОВ**

Учебное пособие

 **УНИВЕРСИТЕТ ИТМО**

Санкт-Петербург

2015

Цуканова О. А. Методология и инструментарий моделирования бизнес-процессов: учебное пособие – СПб.: Университет ИТМО, 2015. – 100 с.

В настоящем учебном пособии рассматриваются основные теоретические положения, связанные с анализом и моделирование бизнес-процессов как на концептуальном уровне (методологии семейства IDEF, диаграммы потоков данных DFD, аппарат сетей Петри), так и на уровне математического моделирования, в частности, моделирование экономических систем с использованием марковских случайных процессов, моделирование систем массового обслуживания.

Учебное пособие разработано в соответствии с программой дисциплины «Методология и инструментарий моделирования бизнес-процессов» и предназначено для студентов всех форм обучения по направлению 38.04.05 «Бизнес-информатика».

Рекомендовано к печати на заседании Ученого совета Гуманитарного факультета, 16.12.2014 г., протокол № 11.



Университет ИТМО – ведущий вуз России в области информационных и фотонных технологий, один из немногих российских вузов, получивших в 2009 году статус национального исследовательского университета. С 2013 года Университет ИТМО – участник программы повышения конкурентоспособности российских университетов среди ведущих мировых научно-образовательных центров, известной как проект «5 в 100». Цель Университета ИТМО – становление исследовательского университета мирового уровня, предпринимательского по типу, ориентированного на интернационализацию всех направлений деятельности.

© Университет ИТМО, 2014

© О. А. Цуканова, 2014

Содержание

Содержание.....	3
Глава 1. Бизнес-процессы и их описание	5
1.1. Понятие бизнес-процесса	5
1.2. Сущность описания и моделирования бизнес-процессов	7
1.3. Способы описания бизнес-процессов.....	8
Глава 2. Концептуальные методологии моделирования бизнес-процессов и данных в системе.....	12
2.1. Семейство стандартов структурного моделирования IDEF	12
2.1. Функциональное моделирование бизнес-процессов в IDEF0	15
2.1. Стандарт IDEF1x	21
2.3. Динамическое моделирование и сети Петри	25
2.4. Методология документирования процессов IDEF3	30
2.5. Методология построения объектно-ориентированных систем IDEF4..	38
2.6. Методология онтологического моделирования	44
2.7. Обзор частично разработанных IDEF методов.....	50
2.7.1. IDEF6 – метод рационального представления процесса проектирования информационных систем	50
2.7.2. IDEF8 – стандарт описания интерфейсов взаимодействия оператора и системы.....	53
2.7.3. IDEF9 - метод анализа условий и ограничений.....	54
2.7.4. IDEF14 - метод проектирования вычислительных сетей.....	56
2.8. Диаграммы потоков данных (DFD).....	57
Глава 3. Математические модели бизнес-процессов	61
3.1. Обзор математических моделей бизнес-процессов	61
3.2. Моделирование экономических систем с использованием марковских случайных процессов.....	63
3.2.1. Основные понятия марковских процессов.....	63
3.2.2. Марковские цепи	64
3.2.3. Непрерывные цепи Маркова.....	68

3.3. Моделирование систем массового обслуживания	78
3.3.1. Компоненты и классификация моделей массового обслуживания .	78
3.3.2. Определение характеристик систем массового обслуживания	82
Литература.....	97

Глава 1. Бизнес-процессы и их описание

1.1. Понятие бизнес-процесса

Бизнес-процесс представляет собой систему последовательных, целенаправленных и регламентированных видов деятельности, в которой посредством управляющего воздействия и с помощью ресурсов входы процесса преобразуются в выходы, результаты процесса, представляющие ценность для потребителей.

Ключевыми свойствами бизнес-процесса является то, что это конечная и взаимосвязанная совокупность действий, определяемая отношениями, мотивами, ограничениями и ресурсами внутри конечного множества субъектов и объектов, объединяющихся в систему ради общих интересов с целью получения конкретного результата, отчуждаемого или потребляемого самой системой.

В стандарте ISO 9000-2001 **процесс** определен как «совокупность взаимосвязанных или взаимодействующих видов деятельности, преобразующих входы в выходы». В литературе встречается также множество определений термина «бизнес-процесс», среди которых можно отметить следующие:

- бизнес-процесс — это последовательность работ, соотнесенная с отдельным видом производственно-хозяйственной деятельности компании и ориентированная на создание новой стоимости, например, на выпуск продукции;
- бизнес-процесс — это упорядоченный по времени набор заданий, выполняемых как людьми, так и информационными системами предприятия, который направлен на достижение заранее известной бизнес-цели за определенное время;
- бизнес-процесс — это набор операций, которые вместе взятые образуют результат, имеющий ценность для потребителя;
- бизнес-процесс — это ряд определенных, поддающихся измерению задач, выполняемых людьми и системами, которые направлены на достижение заранее запланированного результата;
- бизнес-процесс — это устойчивая целенаправленная последовательность исполнения функций, направленная на создание результата, имеющего ценность для потребителя;

- бизнес-процесс — это логичный, последовательный, взаимосвязанный набор мероприятий, который потребляет ресурсы поставщика, создает ценность и выдает результат потребителю;
- бизнес-процесс — это совокупность нескольких связанных между собой процедур либо функций, которые совместно реализуют некоторую задачу бизнеса;
- бизнес-процесс — это полный и динамично координируемый набор совместных бизнес-мероприятий, направленных на принесенные выгоды клиентам;
- бизнес-процесс — это поток работы, у которого есть начало, конец и границы. Он может быть описан с разной степени детализации;
- бизнес-процесс — это последовательность взаимосвязанных действий и событий, направленных на достижение какой-либо цели;
- бизнес-процесс — это набор бизнес-процедур, приводящих к достижению определенного результата.

Бизнес-процессы разделяют на *основные, сопутствующие, вспомогательные, обеспечивающие, процессы управление и процессы развития.*

Основными бизнес-процессами являются процессы, ориентированные на производство товаров или оказание услуги, являющиеся целевыми объектами создания предприятия и обеспечивающие получение заданного результата хозяйственной деятельности.

Сопутствующие бизнес-процессы — это процессы, в результате которых формируются сопутствующие результаты хозяйственной деятельности предприятия.

Вспомогательные и обеспечивающие бизнес-процессы – это процессы, предназначенные для жизнеобеспечения основных и сопутствующих процессов и ориентированные на поддержку их специфических черт.

Бизнес-процессы управления — это процессы, охватывающие весь комплекс функций управления на уровне каждого бизнес-процесса и предприятия в целом.

Бизнес-процессами развития являются процессы совершенствования производимого товара или услуги, процессы развития технологий, процессы модификации оборудования, а также инновационные процессы.

Описание бизнес-процессов проводится с целью их дальнейшего анализа и улучшения. Моделирование бизнес-процессов позволяет проанализировать не только как работает предприятие в целом, как оно взаимодействует со

внешними организациями, заказчиками и поставщиками, но и как организована деятельность на каждом отдельно взятом подразделении, участке, рабочем месте.

1.2. Сущность описания и моделирования бизнес-процессов

Описание бизнес-процессов проводится с целью их дальнейшего анализа и улучшения. Моделирование бизнес-процессов позволяет проанализировать не только как работает предприятие в целом, как оно взаимодействует со внешними организациями, заказчиками и поставщиками, но и как организована деятельность на каждом отдельно взятом подразделении, участке, рабочем месте.

В общем случае модель бизнес-процесса должна давать ответы на следующие вопросы, которые позволят провести всесторонний анализ, взглянуть со всех точек зрения на бизнес-процесс, детализировать его:

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

Бизнес-моделирование — деятельность по выявлению, описанию, анализу существующих бизнес-процессов, а также проектированию новых бизнес-процессов.

Бизнес-моделированием также называют дисциплину и отдельный под-процесс в процессе разработки программного обеспечения, в котором описывается деятельность предприятия и определяются требования к системе (те подпроцессы и операции, которые подлежат автоматизации в разрабатываемой информационной системе).

Под *бизнес-моделью* будем понимать структурированное графическое описание сети процессов и/или функций/операций, связанных с данными, документами, организационными единицами и прочими объектами, отражающими существующую или предполагаемую деятельность организации.

Модель предприятия, всех его бизнес-процессов, сориентированных на конкретную цель, предоставляет возможность его совершенствования. Моделирование бизнес-процессов с максимальной приближенностью к действительности позволяет выбрать и проверить пути улучшения без необходимости проведения реальных экспериментов с предприятием и, таким образом, снизить риск. Бизнес-модель позволяет заранее дать оценку с разных точек зрения. Для предприятия первоочередные требования предъявляются к его функционированию, управлению, эффективности, конечному результату деятельности и степени удовлетворенности клиентов.

1.3. Способы описания бизнес-процессов

1. Текстовый способ. Такой способ представляет собой простое текстовое последовательное описание бизнес-процесса, например: «Отдел продаж составляет договор и согласует его с юридическим отделом». Многие предприятия разработали и используют в своей деятельности регламентирующие документы, часть которых является процессными регламентами и представляет не что иное, как текстовое описание бизнес-процессов.

Очевидно, что для целей анализа и оптимизации деятельности компании данный вариант имеет существенный недостаток, и описание бизнес-процесса в текстовом виде системно рассмотреть и проанализировать фактически невозможно, что связано с устройством человеческого мышления (сложностью чтения как вида деятельности, потребность в образах, «образных закрепителях» и т. п.). Поэтому при использовании текстового описания бизнес-процессов производительность и качество решений по оптимизации

деятельности являются относительно низкими, что особенно сильно проявляется, когда решение принимается группой людей.

2. **Табличный способ.** Табличный способ описания бизнес-процесса является более формализованным и предполагает разбиение бизнес-процесса по ячейкам структурированной таблицы, в которой каждый столбец и строчка имеют некоторое определенное значение.

Пример табличного описания бизнес-процесса

Процесс — закупки. *Владелец* — заместитель коммерческого директора. *Цель процесса* — обеспечение потребности производства материалами и комплектующими. *Краткое описание процесса* — организация обеспечения ТМЦ (товарно-материальных ценностей), хранение и передача их в производство, выбор и оценки поставщиков.

Таблица 1.1

Пример табличного описания бизнес-процесса

№	Субпроцесс	Содержание	Владелец	Участники
1.	Обобщение и уточнение перечня закупаемых ТМЦ	Уточнение перечня закупаемых ТМЦ. Оформление карт разрешения по замене	Начальник отдела внешней комплектации, отдел материально-технического снабжения	Отдел автоматизированного управления производством
2.	Оценка и выбор поставщиков	Подбор альтернативных поставщиков. Сбор данных о поставщике. Оценка поставщиков. Сравнительный анализ поставщиков. Выбор поставщика	Заместитель коммерческого директора	Ведущие инженеры отдела внешней комплектации, отдел материально-технического снабжения
			

Данную таблицу читать проще, из нее легче понять, кто за что отвечает, в какой последовательности в бизнес-процессе выполняются работы, и, соответственно, бизнес-процесс проще проанализировать. Табличная форма описания бизнес-процессов более эффективна по сравнению с текстовой, поэто-

му в настоящее время более широко распространена и применяется для описания бизнес-процессов в приложении к задачам их автоматизации.

Таким образом, при использовании табличного способа на начальном этапе необходимо описать входы и выходы процесса (поставщиков и потребителей), управляющие воздействия (внутренние и внешние) и виды ресурсов (людские и материальные). Для этого составляется «Ведомость определения процесса». Необходимо описать субпроцессы, а также виды сопроводительной документации и риски срыва процесса. Описание бизнес-процесса должно содержать блок-схему и логику процесса. Блок-схема включает процесс, представленный в виде сущностей (прямоугольников произвольной формы), связанных отношениями (стрелками), задающими последовательность выполнения функций процесса. Блок-схема содержит описание следующих атрибутов процесса: владелец процесса, условия начала процесса, заказчик процесса и ожидаемые выходные результаты. При получении выходных данных происходят оценка процесса и анализ фактических показателей. Затем формируются требования к дополнительным ресурсам для улучшения деятельности процесса и определяются виды ресурсов.

Логика процесса может быть представлена в виде таблицы, содержащей столбцы: действие, содержание, ответственный исполнитель и участники. *Действия* — это этапы процесса, причем каждой строке таблицы соответствует свой этап. Совокупность данных этапов определяет алгоритм исполнения процесса, описываемый в виде блок-схемы, где действия представлены в виде прямоугольников, а условные переходы — в виде ромбов. *Содержание* дает представление о действиях на каждом этапе процесса. Это могут быть различные виды документов для данного этапа, а также анализ пройденных этапов (условие перехода), оценки, запросы и пояснения. Каждый этап (действие) имеет своего ответственного исполнителя и участников.

При этом такое представление бизнес-процессов содержит очевидные недостатки. Во-первых, оно не наглядно для пользователей процесса. Во-вторых, каждый раз при внесении изменений в процесс необходимо вручную редактировать соответствующие текстовые документы. В-третьих, при отсутствии унификации представления блок-схем алгоритмов и моделей процессов затруднительно создавать описание, соответствующее общепринятым стандартам оформления бизнес-процессов. Из этого можно сделать вывод о целесообразности перевода уже разработанных таким образом процессов в специализированную инструментальную среду моделирования.

3. **Графический подход.** В настоящее время наибольшее развитие и применение при описании бизнес-процессов получают графические подходы и методы. Признано, что они обладают наибольшей результативностью при решении задач по описанию, анализу и рационализации деятельности предприятия. В случае, когда читателю представляется информация в виде графических образов, значительно возрастают его возможности анализа и принятия решений.

В настоящее время для описания бизнес-процессов существует множество методологий (*IDEF0*, *IDEF3*, *DFD*, *WORKFLOW*, *UML*, *ARIS* и другие) и инструментальных средств (*BPWin*, *ERWin*, *PowerDesigner* и другие).

В структурном и объектно-ориентированном анализе используются средства, моделирующие в форме диаграмм определенного вида деловые процессы и отношения между данными в системе. Этим средствам соответствуют определенные виды системных моделей, наиболее распространены среди них следующие:

- *IDEF (Integrated Definition)* – семейство структурных моделей и соответствующих им диаграмм;
- *DFD (Data Flow Diagrams)* – диаграммы потоков данных;
- *ERD (Entity-Relationship Diagrams)* – диаграммы «сущность-связь»;
- *Workflow* – технология управления потоками работ;
- *BPMN (Business Process Modeling Notation)*;
- средства имитационного моделирования, основанные на математическом аппарате раскрашенных сетей Петри (*Color Petri Nets*, *CPN*);
- объектно-ориентированные методологии на основе унифицированного языка моделирования *UML*;
- интегрированные средства и методологии широкого назначения, например *ARIS*.

Глава 2. Концептуальные методологии моделирования бизнес-процессов и данных в системе

2.1. Семейство стандартов структурного моделирования IDEF

Одними из самых известных и широко используемых методологий в области моделирования бизнес-процессов являются методологии семейства *IDEF*. Семейство *IDEF* появилось в конце 60-х гг. XX в. под названием *SADT* (*Structured Analysis and Design Technique*) В настоящее время оно включает следующие стандарты.

- 1) ***IDEF0*** – методология функционального моделирования. Используется для создания **функциональной модели**, отображающей структуру и функции системы, а также потоки информации и материальных объектов, связывающих эти функции.
- 2) ***IDEF1*** – методология моделирования информационных потоков внутри систем, позволяющая отображать их структуру и взаимосвязи. Методология применяется для построения **информационной модели**, отображающей структуру и содержание информационных потоков, необходимых для поддержки функций системы.
- 3) ***IDEF1X (IDEF1X Extended)*** – методология построения реляционных информационных структур. *IDEF1X* относится к типу методологий «сущность—связь» и, как правило, используется для моделирования реляционных баз данных, имеющих отношение к рассматриваемой системе.
- 4) ***IDEF2*** – методология динамического моделирования развития систем, которая позволяет построить **динамическую модель** меняющихся во времени поведения функций, информации и ресурсов системы. В настоящее время известны алгоритмы и их компьютерные реализации, позволяющие превращать набор статических диаграмм *IDEF0* в динамические модели, построенные на базе «раскрашенных сетей Петри» (*CPN – Color Petri Nets*).
- 5) ***IDEF3*** – методология документирования процессов, происходящих в системе. С помощью *IDEF3* описываются сценарий и последовательность операций для каждого процесса. Функция в диаграмме *IDEF3*

может быть представлена в виде отдельного процесса средствами *IDEF3*.

- 6) ***IDEF4*** – методология построения объектно-ориентированных систем. Средства *IDEF4* позволяют наглядно отображать структуру объектов и заложенные принципы их взаимодействия, позволяя тем самым анализировать и оптимизировать сложные объектно-ориентированные системы.
- 7) ***IDEF5*** – методология онтологического исследования сложных систем. С помощью этой методологии онтология системы описывается при помощи определенного словаря терминов и правил, на основе которых могут быть сформированы достоверные утверждения о состоянии рассматриваемой системы в некоторый момент времени. На основе этих утверждений формируются выводы о дальнейшем развитии системы и производится ее оптимизация.
- 8) ***IDEF6 (Design Rational Capture)*** – метод рационального представления процесса проектирования информационных систем, позволяющий обосновать необходимость проектируемых моделей, выявить причинно-следственные связи и отразить это в итоговой документации системы.
- 9) ***IDEF8 (User Interface Modeling) – Human – System Interaction Design Method*** – метод проектирования взаимодействия пользователей с системами различной природы (не обязательно информационно-вычислительными).
- 10) ***IDEF9 (Business Constraint Discovery Method)*** – метод изучения и анализа бизнес-систем в терминах «ограничений». Ограничения иницируют результат, руководят и ограничивают поведение объектов и агентов (автономных программных модулей) для выполнения целей или намерений системы.
- 11) ***IDEF14 (Network Design Method)*** – метод проектирования вычислительных сетей, позволяющий устанавливать требования, определять сетевые компоненты, анализировать существующие сетевые конфигурации и формулировать желаемые характеристики сети.

Анонсированные корпорацией *KBSI (Knowledge Based System Inc.)* методы *IDEF7 (Information System Audit Method)*, *IDEF10 (Information Artifact Modeling)* и *IDEF12 (Organization Design)* не получили дальнейшего развития.

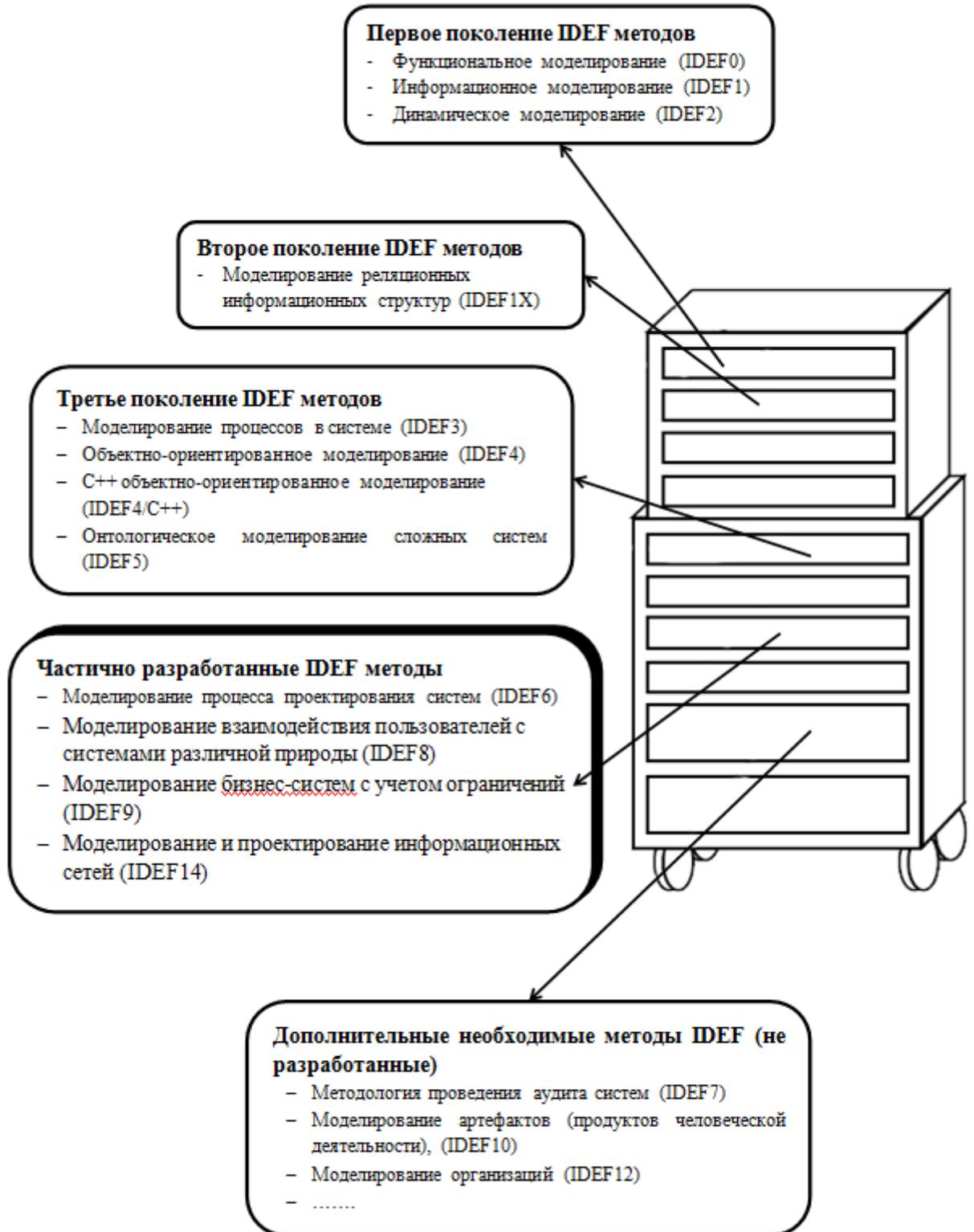


Рис. 2.1. Семейство стандартов структурного моделирования IDEF

С помощью методологий семейства *IDEF* можно эффективно отображать и анализировать модели деятельности широкого спектра сложных систем. К настоящему времени наибольшее распространение и применение

имеют методологии *IDEF0* и *IDEF1 (IDEF1X)*, получившие в США статус федеральных стандартов.

2.1. Функциональное моделирование бизнес-процессов в *IDEF0*

IDEF0 может быть использована для моделирования широкого класса систем. Для новых систем применение *IDEF0* направлено на определение требований и указание функций для последующей разработки системы, отвечающей поставленным требованиям и реализующей выделенные функции. Применительно к уже существующим системам *IDEF0* может быть использована для анализа функций, выполняемых системой, и отображения механизмов, посредством которых эти функции выполняются. Результатом применения *IDEF0* к некоторой системе является модель этой системы, состоящая из иерархически упорядоченного набора диаграмм, текста документации и словарей, связанных друг с другом с помощью перекрестных ссылок. Двумя наиболее важными компонентами, из которых строятся диаграммы *IDEF0*, являются бизнес-функции, или работы (блоки), и данные, или объекты (дуги), связывающие между собой работы и отображающие взаимодействия и взаимосвязи между ними.

Синтаксис графического языка *IDEF0*

Набор структурных компонентов языка, их характеристики и правила, определяющие связи между компонентами, представляют собой синтаксис языка. Компоненты синтаксиса *IDEF0* – *блоки, стрелки, диаграммы и правила*.

Блоки представляют функции, определяемые как деятельность, процесс, операция, действие или преобразование. Блок описывает функцию. Внутри каждого блока помещается его имя и номер. Имя должно быть активным глаголом или глагольным оборотом, описывающим функцию. Номер блока размещается в правом нижнем углу. Номера блоков используются для их идентификации на диаграмме и в соответствующем тексте.

РАЗРАБОТАТЬ МОДЕЛЬ 1	<ul style="list-style-type: none"> • Имя функции – глагол или глагольный оборот • Показан номер блока
--------------------------------	---

Требования к блокам в IDEF0 следующие:

- 1) размеры блоков должны быть достаточными для того, чтобы включить имя блока;
- 2) блоки должны быть прямоугольными, с прямыми углами;
- 3) блоки должны быть нарисованы сплошными линиями.

Стрелки представляют данные или материальные объекты, связанные с функциями.

Стрелка формируется из одного или более отрезков прямых и наконечника на одном конце. Сегменты стрелок могут быть прямыми или ломаными. Стрелки не представляют поток или последовательность событий, как в традиционных блок-схемах потоков или процессов. Они лишь показывают, какие данные или материальные объекты должны поступить на вход функции для того, чтобы эта функция могла выполняться.

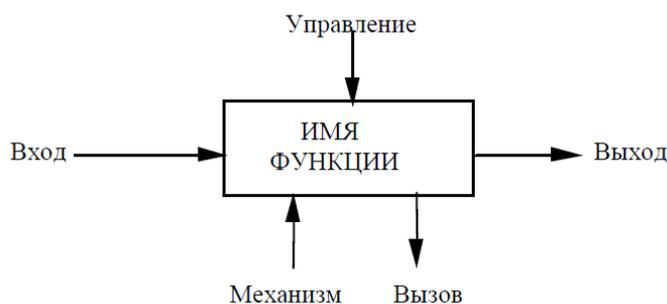
Требования к стрелкам в IDEF0 следующие:

- 1) ломаные стрелки изменяют направление только под углом 90 градусов;
- 2) стрелки должны быть нарисованы сплошными линиями различной толщины;
- 3) стрелки могут состоять только из вертикальных или горизонтальных отрезков, отрезки, направленные по диагонали, не допускаются;
- 4) концы стрелок должны касаться внешней границы функционального блока, но не должны пересекать ее;
- 5) стрелки должны присоединяться к блоку на его сторонах, присоединение в углах не допускается;
- 6) каждая стрелка должна быть помечена существительным или боротом существительного (например, «отчет об испытаниях», «конструкция детали», «бюджет», «конструкторские требования» и др.).

Стрелки, представляя множества объектов, в зависимости от того, в какую грань блока (прямоугольника работы) они входят или из какой грани выходят, делятся на пять видов:

- *входа* (входят в левую грань работы) – изображают данные или объекты, изменяемые в ходе выполнения работы;
- *управления* (входят в верхнюю грань работы) – изображают правила и ограничения, согласно которым выполняется работа;
- *выхода* (выходят из правой грани работы) – изображают данные или объекты, появляющиеся в результате выполнения работы;
- *механизма* (входят в нижнюю грань работы) – изображают ресурсы, необходимые для выполнения работы, но не изменяющиеся в процессе работы (например, оборудование, людские ресурсы и т.д.);
- *вызова* (выходят из нижней грани работы) – изображают связи между разными диаграммами или моделями, указывая на некоторую диаграмму, где данная работа рассмотрена более подробно.

Входные дуги изображают объекты, используемые и преобразуемые функциями. *Управленческие дуги* представляют информацию, управляющую действиями функций. *Выходные дуги* изображают объекты, в которые преобразуются входы. *Дуги механизмов IDEF0* изображают физические аспекты функции (склады, людей, организации, приборы). Таким образом, стороны блока графически сортируют объекты, изображаемые касающимися блока дугами.



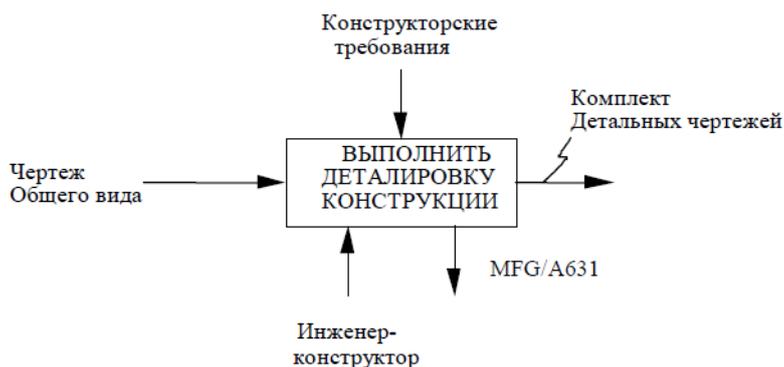
Семантика языка IDEF0

Семантика определяет содержание (значение) синтаксических компонентов языка и способствует правильности их интерпретации. Интерпретация устанавливает соответствие между блоками и стрелками с одной стороны и функциями и их интерфейсами – с другой.

Семантические правила блоков и стрелок:

- 1) имя блока должно быть активным глаголом или глагольным оборотом;
- 2) каждая сторона функционального блока должна иметь стандартное отношение блок/стрелки:

- входные стрелки должны связываться с левой стороной блока;
 - управляющие стрелки должны связываться с верхней стороной блока;
 - выходные стрелки должны связываться с правой стороной блока;
 - стрелки механизма (кроме стрелок вызова) должны указывать вверх и подключаться к нижней стороне блока;
 - стрелки вызова механизма должны указывать вниз, подключаться к нижней стороне блока, и помечаться ссылкой на вызываемый блок;
- 3) сегменты стрелок, за исключением стрелок вызова, должны помечаться существительным или оборотом существительного, если только единственная метка стрелки несомненно не относится к стрелке в целом;
 - 4) чтобы связать стрелку с меткой, следует использовать "тильду";
 - 5) в метках стрелок не должны использоваться следующие термины: функция, вход, управление, выход, механизм, вызов.



Первая диаграмма в иерархии диаграмм IDEF0 всегда изображает функционирование системы в целом. Такие диаграммы называются *контекстными*. В контекст входят описание цели моделирования, области (описание того, что будет рассматриваться как компонент системы, а что – как внешнее воздействие) и точка зрения (позиция, с которой будет строиться модель). Обычно в качестве точки зрения выбирается точка зрения лица или объекта, ответственного за работу моделируемой системы в целом. После описания контекста строят следующие диаграммы в иерархии. Каждая последующая диаграмма является более подробным описанием (декомпозицией) одной из работ на вышестоящей диаграмме (рис. 2.2). Описание каждой подсистемы проводится аналитиком совместно с экспертом предметной области. Обычно экспертом является человек, отвечающий за эту подсистему, и поэтому дос-

конально знающий все ее функции. Таким образом, вся система разбивается на подсистемы до нужного уровня детализации, и получается модель, аппроксимирующая систему с заданным уровнем точности. Получив модель, адекватно отображающую текущие бизнес-процессы (так называемую модель *AS-IS*), аналитик может увидеть все наиболее уязвимые места системы. После этого с учетом выявленных недостатков можно строить модель новой организации бизнес-процессов (модель *TO-BE*).

IDEF0 модель имеет единственную цель и единственный субъект. Цель модели - получение ответов на определенную совокупность вопросов. Субъект — это сама система. Методология IDEF0 требует, чтобы создаваемая модель системы рассматривалась всегда с одной и той же позиции, или точки зрения. После определения точки зрения, с которой описывается модель, создается список данных, а потом список функций.

Каждый блок диаграммы IDEF0-модели может быть детализирован на другой диаграмме. Поскольку каждый блок понимается как отдельный, полностью определенный объект, разделение такого объекта на его структурные части (блоки и дуги, составляющие диаграмму) называется *декомпозицией*. Декомпозиция формирует границы, и каждый блок в IDEF0 рассматривается как формальная граница некоторой части описываемой системы, т.е. блок и касающиеся его дуги определяют точную границу диаграммы, представляющей декомпозицию этого блока. Эта диаграмма, называемая *диаграммой-потомком*, описывает все, связанное с этим блоком и его дугами, и не описывает ничего вне этой границы.

Декомпозируемый блок называется *родительским блоком*, а содержащая его диаграмма – *родительской диаграммой*.

IDEF0 требует, чтобы все внешние дуги (ведущие к краю страницы) диаграммы были согласованы с дугами, образующими границу этой диаграммы, т.е. диаграмма должна быть «состыкована» со своей родительской диаграммой посредством согласования по числу и наименованию дуг.

В методологии принята схема кодирования дуг «ICOM», которая получила название по первым буквам английских эквивалентов слов: вход (Input), управление (Control), выход (Output), механизм (Mechanism). При построении диаграммы следующего уровня дуги, касающиеся декомпозируемого блока, используются в качестве источников и приемников для дуг, которые создаются на новой диаграмме. После завершения диаграммы ее внешние дуги

стыкуются с родительской диаграммой для обеспечения согласованности. Стыковка осуществляется посредством присваивания кодов ICOM внешним дугам новой диаграммы. Таким образом, IDEF0-диаграмма составлена из блоков, связанных дугами, которые определяют, как блоки влияют друг на друга. Дуги на диаграммах изображают интерфейсы между функциями системы, а также между системой и ее окружающей средой.

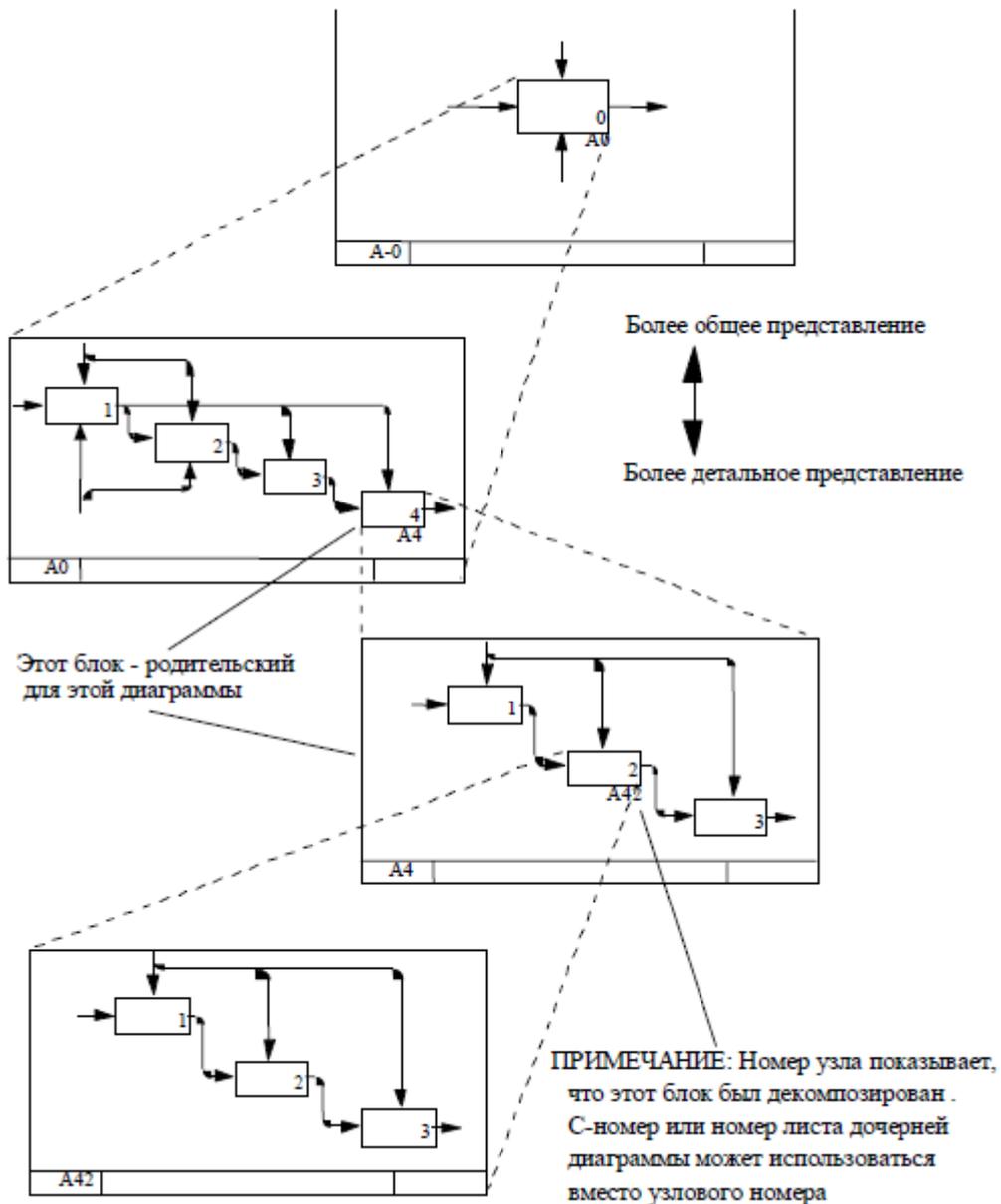


Рис. 2.2. Декомпозиция диаграмм в IDEF0

Существуют некоторые критерии для определения момента завершения моделирования:

- блок содержит достаточно деталей;

- необходимо изменить уровень абстракции, чтобы достичь большей детализации блока;
- необходимо изменить точку зрения, чтобы детализировать блок;
- блок очень похож на другой блок той же модели;
- блок очень похож на блок другой модели;
- блок представляет тривиальную функцию.

2.1. Стандарт IDEF1x

IDEF1X является методом для разработки реляционных баз данных и использует условный синтаксис, специально разработанный для удобного построения концептуальной схемы. Использование метода IDEF1X наиболее целесообразно для построения логической структуры базы данных после того как все информационные ресурсы исследованы и решение о внедрении реляционной базы данных, как части корпоративной информационной системы, было принято. Средства моделирования IDEF1X специально разработаны для построения реляционных информационных систем, и если существует необходимость проектирования другой системы, например, объектно-ориентированной, то лучше избрать другие методы моделирования.

Существует несколько очевидных причин, по которым IDEF1X не следует применять в случае построения нереляционных систем. Во-первых, IDEF1X требует от проектировщика определить ключевые атрибуты, для того чтобы отличить одну сущность от другой, в то время как объектно-ориентированные системы не требуют задания ключевых ключей, в целях идентификации объектов. Во-вторых, в тех случаях, когда более чем один атрибут является однозначно идентифицирующим сущность, проектировщик должен определить один из этих атрибутов первичным ключом, а все остальные вторичными. И, таким образом, построенная проектировщиком IDEF1X-модель и переданная для окончательной реализации программисту является некорректной для применения методов объектно-ориентированной реализации, и предназначена для построения реляционной системы.

Концепция и семантика IDEF1X

Сущность в IDEF1X описывает собой совокупность или набор экземпляров похожих по свойствам, но однозначно отличаемых друг от друга по одному или нескольким признакам. Каждый экземпляр является реализацией

сущности. Таким образом, сущность в IDEF1X описывает конкретный набор экземпляров реального мира, в отличие от сущности в IDEF1, которая представляет собой абстрактный набор информационных отображений реального мира.

Примером сущности IDEF1X может быть сущность «Сотрудник», которая представляет собой всех сотрудников предприятия, а один из них, например, Иванов Петр Сергеевич, является конкретной реализацией этой сущности. В примере (рис. 2.3), каждый экземпляр сущности «Сотрудник» содержит следующую информацию: ID сотрудника, имя сотрудника, адрес сотрудника и т.п. В IDEF1X модели эти свойства называются атрибутами сущности. Каждый атрибут содержит только часть информации о сущности.

Связи в IDEF1X представляют собой ссылки, соединения и ассоциации между сущностями. Связи показывают как соотносятся сущности между собой.

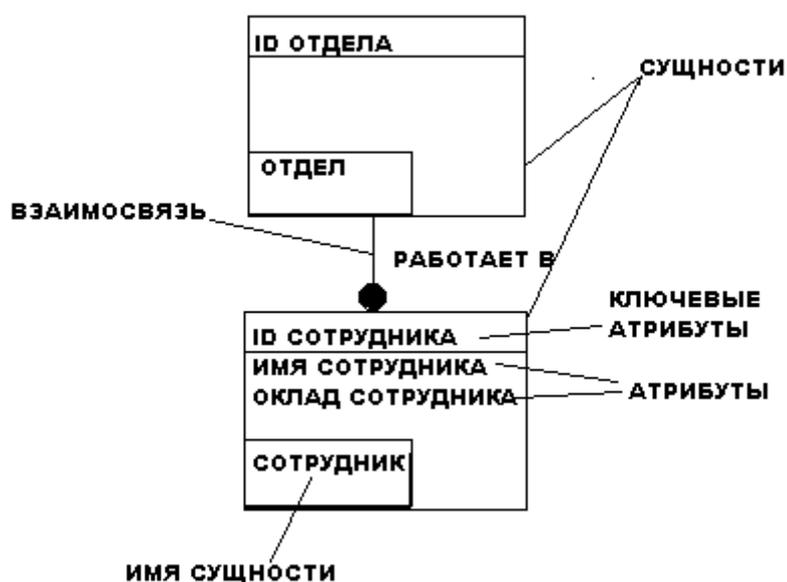


Рис. 2.3. Диаграмма связи между Сотрудником и Отделом

В приведенном примере взаимосвязи между сущностями соответствуют схеме *один ко многим*. Это означает, что один экземпляр первой сущности связан с несколькими экземплярами второй сущности. Причем первая сущность называется родительской, а вторая - дочерней.

Отношения *многие ко многим* обычно используются на начальной стадии разработки диаграммы, например, в диаграмме зависимости сущностей и отображаются в IDEF1X в виде сплошной линии с точками на обоих концах.

Так как отношения многие ко многим могут скрыть другие бизнес правила или ограничения, они должны быть полностью исследованы на одном из этапов моделирования.

Сущность описывается в диаграмме IDEF1X графическим объектом в виде прямоугольника. Верхняя часть называется ключевой областью, а нижняя часть областью данных. Ключевая область объекта «Сотрудник» содержит поле «Уникальный идентификатор сотрудника», в области данных находятся поля «Имя сотрудника», «Адрес сотрудника», «Телефон сотрудника» и т.д.

Ключевая область содержит *первичный ключ* для сущности. Первичный ключ - это набор атрибутов, выбранных для идентификации уникальных экземпляров сущности. Атрибуты первичного ключа располагаются над линией в ключевой области. Неключевые атрибуты располагаются под чертой, в области данных.

В качестве *первичных ключей* могут быть использованы несколько атрибутов или групп атрибутов. Ни одна из частей ключа не может быть NULL, не заполненной или отсутствующей.

В примере атрибут «ID сотрудника» является потенциальным ключом, так как он уникален для всех экземпляров сущности «Сотрудник». Также комбинация атрибутов «имя сотрудника» и «дата рождения сотрудника» может оказаться удачной и стать искомым потенциальным ключом.

При выборе первичного ключа для сущности разработчики модели часто используют дополнительный (суррогатный) ключ, то есть произвольный номер, который уникальным образом определяет запись в сущности. Атрибут «Номер сотрудника» является примером суррогатного ключа. Суррогатный ключ лучше всего подходит на роль первичного ключа потому, что является коротким и быстрее всего идентифицирует экземпляры в объекте. К тому же суррогатные ключи могут автоматически генерироваться системой так, чтобы нумерация была сплошной.

Потенциальные ключи, которые не выбраны первичными, могут быть использованы в качестве вторичных или альтернативных ключей. С помощью альтернативных ключей часто отображают различные индексы доступа к данным в конечной реализации реляционной базы.

Если сущности в IDEF1X диаграмме связаны, связь передает ключ (или набор ключевых атрибутов) дочерней сущности. Эти атрибуты называются внешними ключами. Внешние ключи определяются как атрибуты первичных

ключей родительского объекта, переданные дочернему объекту через их связь. Передаваемые атрибуты называются мигрирующими.

При разработке модели приходится сталкиваться с сущностями, уникальность которых зависит от значений атрибута внешнего ключа. Для этих сущностей (для уникального определения каждой сущности) внешний ключ должен быть частью первичного ключа дочернего объекта.

Дочерняя сущность, уникальность которой зависит от атрибута внешнего ключа, называется *зависимой сущностью*. В примере (рис. 2.3) сущность «Сотрудник» является зависимой сущностью потому, что его идентификация зависит от сущности «Отдел». В обозначениях IDEF1X зависимые сущности представлены в виде закругленных прямоугольников.

Зависимые сущности далее классифицируются на сущности, которые не могут существовать без родительской сущности и сущности, которые не могут быть идентифицированы без использования ключа родителя (сущности, зависящие от идентификации). Сущность «Сотрудник» принадлежит ко второму типу зависимых сущностей, так как сотрудники могут существовать и без отдела.

Сущности, независимые при идентификации от других объектов в модели, называются *независимыми сущностями*. В вышеописанном примере сущность «Отдел» можно считать независимой. В IDEF1X независимые сущности представлены в виде прямоугольников.

В IDEF1X концепция зависимых и независимых сущностей усиливается типом взаимосвязей между двумя сущностями. Если вы хотите, чтобы внешний ключ передавался в дочернюю сущность (и, в результате, создавал зависимую сущность), то можете создать идентифицирующую связь между родительской и дочерней сущностью. Идентифицирующие взаимосвязи обозначаются сплошной линией между сущностями.

Неидентифицирующие связи, являющиеся уникальными для IDEF1X, также связывают родительскую сущность с дочерней. Неидентифицирующие связи используются для отображения другого типа передачи атрибутов внешних ключей - передача в область данных дочерней сущности (под линией).

Неидентифицирующие связи отображаются пунктирной линией между объектами. Так как переданные ключи в неидентифицирующей связи не являются составной частью первичного ключа дочерней сущности, то этот вид связи не проявляется ни в одной идентифицирующей зависимости. В этом

случае и «Отдел», и «Сотрудник» рассматриваются как независимые сущности.

Основным преимуществом IDEF1X, по сравнению с другими многочисленными методами разработки реляционных баз данных является жесткая и строгая стандартизация моделирования. Установленные стандарты позволяют избежать различной трактовки построенной модели.

2.3. Динамическое моделирование и сети Петри

Методология IDEF2 реализует динамическое моделирование системы. В данной методологии модель разбивается на четыре подмодели:

- подмодель возможностей, которая описывает их инициаторов;
- подмодель потока сущностей, которая определяет их трансформацию;
- подмодель распределения ресурсов, необходимых для осуществления переходов между состояниями;
- подмодель системы, которая описывает внешние взаимодействия.

Методология предполагает, что набор подмоделей может быть переведен в динамическую модель.

В связи с весьма серьезными сложностями анализа динамических систем от этого стандарта практически отказались, и его развитие приостановилось на самом начальном этапе. В настоящее время известны алгоритмы и их компьютерные реализации, позволяющие превращать набор статических диаграмм IDEF0 в динамические модели, построенные на базе «**раскрашенных сетей Петри**» (*CPN – Color Petri Nets*).

Классические сети Петри ввел Карл Адам Петри в 60-х гг. XX в. С тех пор их использовали для моделирования и анализа самых разных систем с приложениями от протоколов, аппаратных средств и внедренных систем до гибких производственных систем, пользовательского взаимодействия и бизнес-процессов.

Основные определения сетей Петри

Сети Петри используются для моделирования параллельных процессов: для моделирования компонентов компьютера, параллельных вычислений, в робототехнике и даже для описания музыкальных структур. Вообще, сети

Петри используют для нахождения дефектов в проекте системы, хотя имеют и многие другие применения. Они обладают многими свойствами блок-схем и конечных автоматов.

Сети Петри были разработаны и используются для моделирования параллельных и асинхронных систем. При моделировании в сетях Петри позиции символизируют какое-либо состояние системы, а переход символизируют какие-то действия, происходящие в системе. Система, находясь в каком-то состоянии, может порождать определенные действия, и наоборот, выполнение какого-то действия переводит систему из одного состояния в другое.

Сетью Петри называется совокупность множеств $S = \{P, T, I, O\}$, где:

P – конечное множество, элементы которого называются позициями;

T – конечное множество, элементы которого называются переходами,

$P \cap T = \emptyset$;

I – множество входных функций, $I: T \rightarrow P$;

O – множество выходных функций, $O: T \rightarrow P$.

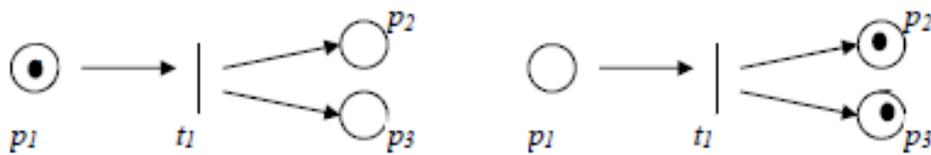
Сеть Петри представляет собой ориентированный граф с вершинами двух типов (*позициями и переходами*), в котором дугами могут соединяться только вершины различных типов. В позиции сети помещаются специальные маркеры («фишки»), перемещение которых и отображает динамику моделируемой системы. Изменение маркировки (движение маркеров) происходит в результате выполнения (срабатывания) перехода на основе соответствующего внешнего события. Точнее, переход срабатывает, если во всех его входных позициях имеются маркеры и происходит соответствующее переходу событие. При этом из каждой входной позиции срабатываемого перехода маркер удаляется, а в каждую выходную позицию — заносится.

Сеть Петри называется *маркированной*, если существует функция μ , называемая *маркировкой (разметкой) сети*, которая ставит в соответствие неотрицательное целое число каждому элементу множества P . Если p – позиция, то $\mu(p)$ называется *разметкой позиции* p . Таким образом, маркированная сеть Петри задается пятеркой $S = \{P, T, I, O, \mu\}$, где μ – целочисленный вектор $\mu = (\mu_1, \mu_2, \dots, \mu_n)$, $n = |P|$, $\mu_i = \mu(p_i)$, $i = 1 \dots n$.

Разметка множества на графе указывается с помощью черных точек, называемых метками (фишками), помещенных в кружки, которые обозначают позиции. Количество меток можно также указывать числом, записанном в кружке. Если кружок позиции p пуст, это означает, что в p меток нет. Маркировка сети Петри аналогична состоянию конечного автомата.

При моделировании гибких производственных систем позиции отражают отдельные операции производственного процесса (например: транспортировка заготовки к конвейеру, передвижение заготовки к станку конвейером, обработку детали) или состояния компонентов гибкой производственной системы (например: робота, конвейера, станка). Наличие метки в одной из позиций соответствует состоянию выполнения некоторой из технологических операций либо состоянию, в котором пребывают некоторые из компонентов гибкой производственной системы.

Переходы соответствуют событиям, отображающим начало или завершение моделируемых операций. Например, переход интерпретируется как событие, связанное с завершением операции транспортирования заготовки роботом и ее установки на конвейере, а также с началом операции перемещения заготовки конвейером к станку.



$$S_{\mu} = \{P, T, I, O, \mu\},$$

$$P = \{p_1; p_2; p_3\}; T = \{t_1\}; I(t_1) = \{p_1\}; O(t_1) = \{p_2; p_3\}; \mu = (1; 0; 0); \mu_1 = (0; 1; 1)$$

Рис. 2.4. Пример сети Петри S_{μ}

Переход называется *разрешенным*, если каждая из его входных позиций имеет число меток не меньше, чем число дуг из позиции в переход. Переход срабатывает только в том случае, когда он разрешен.

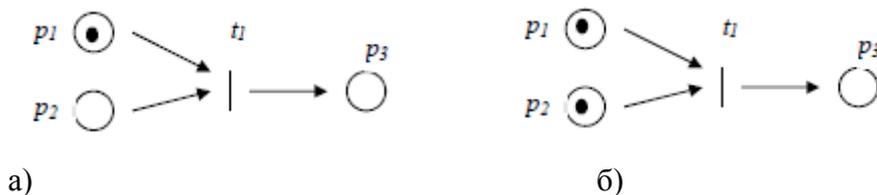


Рис. 2.5. Пример неразрешенного (а) и разрешенного (б) переходов

На рис. 2.5 (а) переход t_1 не может сработать, поскольку в p_2 нет метки. Он будет разрешен, если p_1 и p_2 будут иметь хотя бы по одной метке (б).

Кратные метки необходимы для кратных входных дуг. Метки во входной позиции, которые разрешают переход, называются его *разрешающими метками*.

В результате срабатывания во всех входных позициях перехода число меток уменьшается на величину, равную числу дуг, выходящих из соответствующей позиции в переход, а в выходные позиции данного перехода добавляется число меток, равное числу дуг, исходящих из перехода в соответствующую выходную позицию. Заметим, что срабатывание перехода и изменение маркировок всех связанных с данным переходом позиций осуществляется мгновенно. Если в некоторый момент времени разрешен более чем один переход, первым может сработать любой из них.

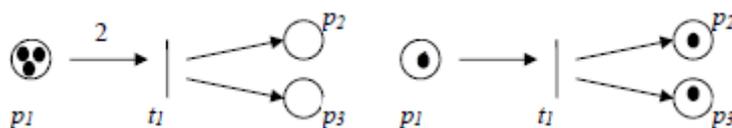


Рис. 2.6. Пример сети Петри с разрешенным переходом

Маркировка μ_n называется *достижимой* из маркировки μ_1 , если существует последовательность срабатывающих переходов, переводящая сеть из маркировки μ_n в μ_1 .

Переход t_j называется *достижимым* из маркировки μ_1 , если существует такая маркировка μ_n , достижимая из μ_1 , при которой происходит срабатывание перехода t_j .

Для доказательства недостижимости t_j из маркировки μ_1 необходимо перебрать все последовательности переходов, срабатывание которых переводит сеть из μ_n в μ_1 .

Сеть Петри называется *живой*, если для любого текущего состояния существует такая последовательность переходов с началом в текущем состоянии, что любой заданный переход может сработать. Доказав, что сеть является живой, можно гарантировать, что в соответствующей системе выполнимы все элементарные действия процессов при их развитии.

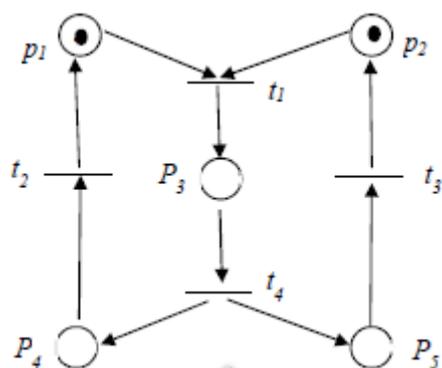


Рис. 2.7. Примеры живой сети

Сеть, не являющаяся живой, называется **тупиковой**. Частичный тупик – такая маркировка, когда один или более переходов могут никогда не сработать. Сеть Петри находится в тупике, если имеется маркировка, в котором ни один из переходов не может сработать.

Сеть Петри называется **безопасной**, если каждая позиция содержит не более одной метки. В безопасной сети Петри в каждой позиции имеется либо одна метка, либо меток нет вообще. Наличие метки может означать протекание процесса, а ее отсутствие – сигнал о его остановке, поэтому для большинства систем контроля моделируют именно безопасную сеть Петри.

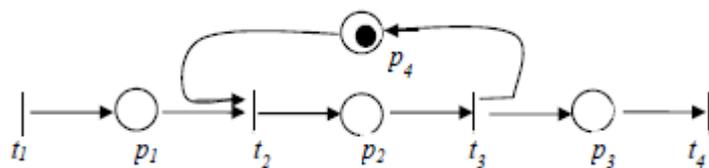
Сеть Петри называется **k -ограниченной**, если количество меток в каждой позиции не превышает некоторое целое число k . Ограниченная сеть Петри предоставляет возможность контроля проблемы переполнения. Очевидно, что безопасная цепь Петри – ограниченная, $k=1$.

Сеть Петри называется **консервативной**, если сумма меток во всех позициях постоянна. Консервативная сеть является ограниченной. У консервативной сети количество меток на входе каждого перехода равно количеству меток на выходе. Если метки описывают ресурсы, то консервативная сеть Петри гарантирует, что никакой ресурс не будет ни создан, ни утерян.

Пример построения сетей Петри

Построить сеть Петри, моделирующую работу рабочей станции, обслуживающей группу пользователей. Пользователь присылает заявку на обработку задания. Если станция свободна, она начинает обработку задания. После выполнения задания станция передает обслуженную заявку, освобождая

ется и либо начинает обрабатывать новую заявку (если заявка поступила), либо ждет поступления новой заявки.



t_1 – поступила заявка на обработку;

t_2 – задание начинает обрабатываться;

t_3 – конец обработки задания;

t_4 – передача выполненной заявки;

p_1 – задание ждет освобождения станции;

p_2 – задание обрабатывается;

p_3 – задание ожидает очереди на выход;

p_4 – рабочая станция свободна.

Позиция p_4 показывает, свободна ли рабочая станция. Наличие метки в позиции указывает на то, что станция свободна. Как только задание начинает обрабатываться, срабатывает переход t_2 , и маркировка позиции обнуляется. После окончания обработки запускается переход t_3 и позиция p_4 вновь получает метку. Таким образом, пока не сработает переход t_3 , новая заявка не может быть обработана.

2.4. Методология документирования процессов IDEF3

Стандарт IDEF3 – это методология описания процессов, рассматривающая последовательность выполнения и причинно-следственные связи между ситуациями и событиями для структурного представления знаний о системе. При помощи IDEF3 описывают логику выполнения работ, очередность их запуска и завершения, т.е. IDEF3 предоставляет инструмент моделирования сценариев действий сотрудников организации, отделов, цехов и т.п., например, порядок обработки заказа или события, на которые необходимо реагировать за конечное время, выполнение действий по производству товара и т.д.

IDEF3 как инструмент моделирования фиксирует следующую информацию о процессе:

- объекты, которые участвуют при выполнении сценария;
- роли, которые выполняют эти объекты, например, агент, транспорт;

- отношения между работами в ходе выполнения сценария процесса;
- состояния и изменения, которым подвергаются объекты;
- время выполнения и контрольные точки синхронизации работ;
- ресурсы, которые необходимы для выполнения работ.

Средства документирования и моделирования IDEF3 позволяют выполнять следующие задачи:

- документировать имеющиеся данные о технологии выполнения процесса, выявленные, например, во время опроса специалистов предметной области, ответственных за организацию рассматриваемого процесса или участвующих в нем;
- анализировать существующие процессы и разрабатывать новые;
- определять и анализировать точки влияния потоков сопутствующего документооборота на сценарий технологических процессов;
- определять ситуации, в которых требуется принятие решения, влияющего на жизненный цикл процесса, например, изменение конструктивных, технологических или эксплуатационных свойств конечного продукта;
- содействовать принятию оптимальных решений при реорганизации процессов;
- разрабатывать имитационные модели технологических процессов по принципу «как будет, если...».

Таким образом, IDEF3 – это методология, способная фиксировать и структурировать описание работы системы. При этом сбор сведений может производиться из многих источников, что позволяет зафиксировать информацию от экспертов о поведении системы.

Основные элементы IDEF3-диаграмм

Основные элементы IDEF3-диаграмм представлены на рис. 2.8.

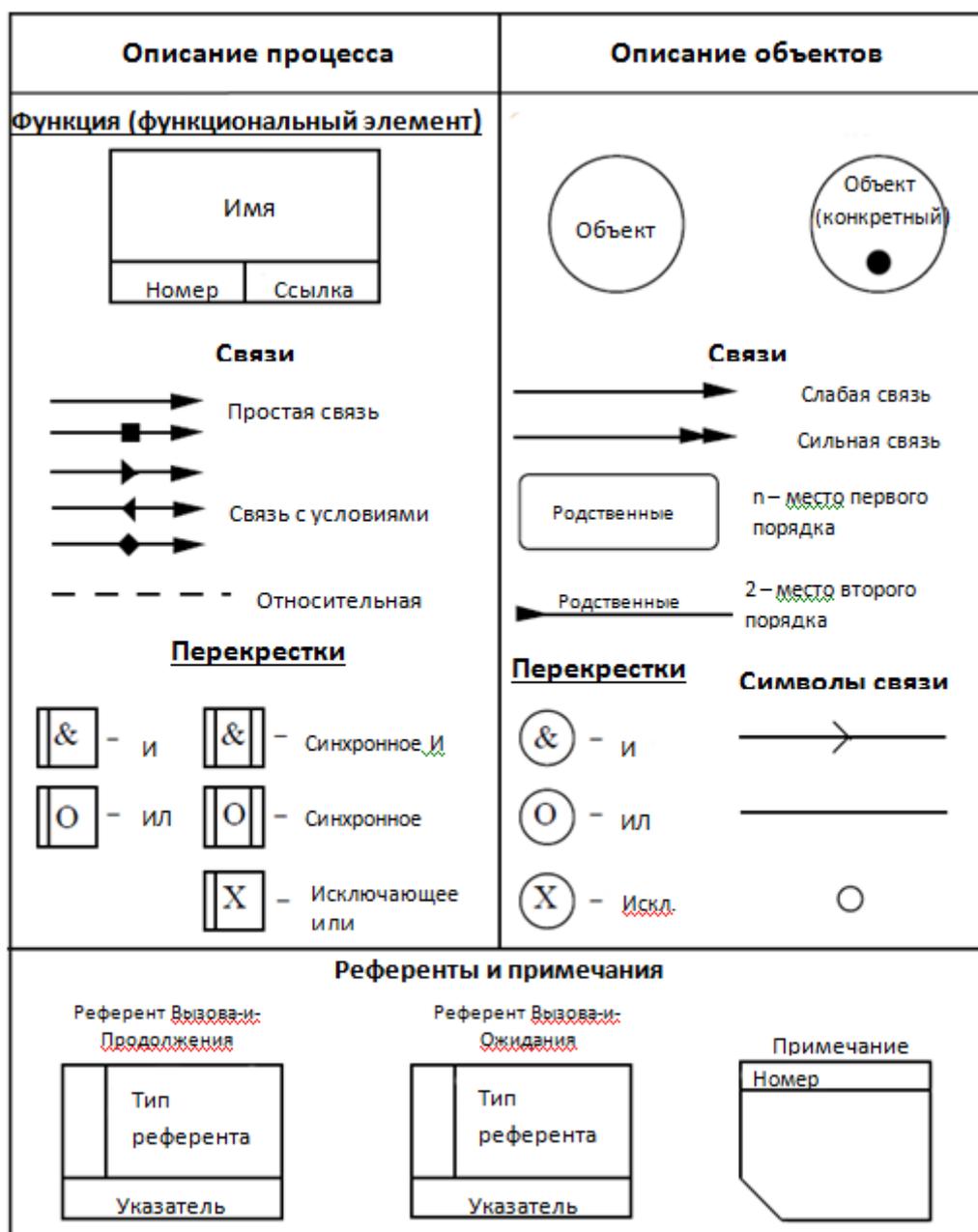


Рис. 2.8. Элементы IDEF3-диаграмм

Функциональный элемент (UOB). Описание процесса представляет всевозможные ситуации (процессы, функции, действия, акты, события, сценарии, процедуры, операции или решения), которые могут происходить в моделируемой системе в логических и временных отношениях. Каждый процесс представлен полем, отображающим название процесса. Номер идентификатора процесса назначается последовательно. В правом нижнем углу UOB-элемента располагается ссылка и используется для указания ссылок либо на элементы из функциональной модели IDEF0, либо для указания на от-

дела или конкретных исполнителей, которые будут выполнять указанную работу.

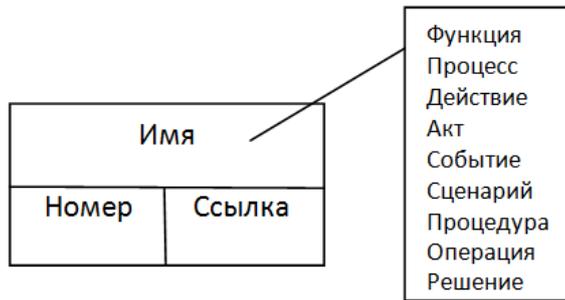


Рис. 2.9. Синтаксис UOB-элемента

Элемент связи необходим для организации отношений между элементами диаграммы и описания динамики происходящих процессов. Связи используются прежде всего для обозначения отношений между функциональными элементами UOB, отображения временной последовательности выполнения сценариев в диаграммах описания процесса. Данные элементы используются в основном для обозначения существенных связей между UOB.

Связи между функциональными блоками могут быть: временные, логические, причинно-следственные, природные и обычные. В подавляющем большинстве случаев используются связи, отражающие простое временное отношение между блоками. Существует два основных типа связей, используемых в IDEF3 схемах: связи приоритета (старшинства) и относительные (прерывистые) связи. Символы, которые представляют каждый вид, показаны на рис. 2.9.

Связи простой очередности демонстрируют временной приоритет отношений между функциональными блоками UOB. Они являются наиболее широко используемыми связями и обозначаются сплошной стрелкой, а иногда дополнительным маркером, прикрепленным к стволу стрелки. Очередность подключения UOB блоков с простой очередностью показана на рис. 2.10.

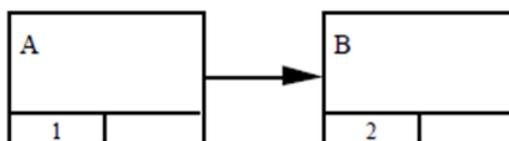


Рис. 2.10. Порядок подключения UOB блоков с простой очередностью

Относительные (прерывистые) связи не несут никакой определенной семантики. По этой причине их часто называют пользовательскими связями. Этот тип ссылок подчеркивает существование (возможно, ограничивающие) отношения между двумя UOBs. Например, связь на рис. 2.11 может означать ограничение между блоками «Подписать расписание» «Получить расписание», которое свидетельствует о том, что нельзя утверждать собственное расписание. Точный характер отношения указывают в документе Разработки.

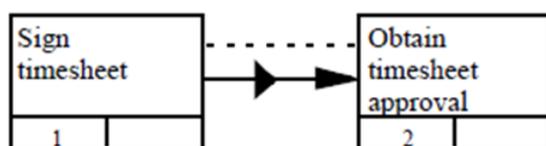


Рис. 2.11. Пример относительной связи между UOB блоками

Перекрестки используются для отображения логики отношений между множеством событий и временной синхронизации активизации элементов IDEF3-диаграмм. Различают перекрестки для слияния (*Fan-in Junction*) и разветвления (*Fan-out Junction*) стрелок. Перекресток не может использоваться одновременно для слияния и для разветвления. При внесении перекрестка в диаграмму необходимо указать его тип. Тип перекрестка определяет логику и временные параметры отношений между элементами диаграммы. Все перекрестки на диаграмме нумеруются, каждый номер имеет префикс «J». Тип перекрестка обозначается внутри элемента: & - логический И; O – логический ИЛИ; X – логический перекресток неэквивалентности.

Стандарт IDEF3 предусматривает разделение перекрестков типа & и O на *синхронные* и *асинхронные*. Это разделение позволяет учитывать в диаграммах описания процессов синхронизацию времени активизации.

Методология IDEF3 использует пять логических типов для моделирования возможных последствий действий в сценарии.

Таблица 2.1

Логические типы

Обозначение	Наименование	Смысл в случае слияния стрелок (<i>Fan-in Junction</i>)	Смысл в случае разветвления стрелок (<i>Fan-out Junction</i>)
	<i>Asynchronous AND</i>	Все предшествующие процесс должны быть завершены	Все следующие процессы должны быть запущены
	<i>Asynchronous OR</i>	Один или несколько предшествующих процессов должны быть завершены	Один или несколько следующих процессов должны быть запущены
	<i>XOR (Exclusive OR)</i>	Только один предшествующий процесс завершен	Только один следующий процесс запускается
	<i>Synchronous AND</i>	Все предшествующие процессы завершены одновременно	Все следующие процессы запускаются одновременно
	<i>Synchronous OR</i>	Один или несколько предшествующих процессов завершаются одновременно	Один или несколько следующих процессов запускаются одновременно

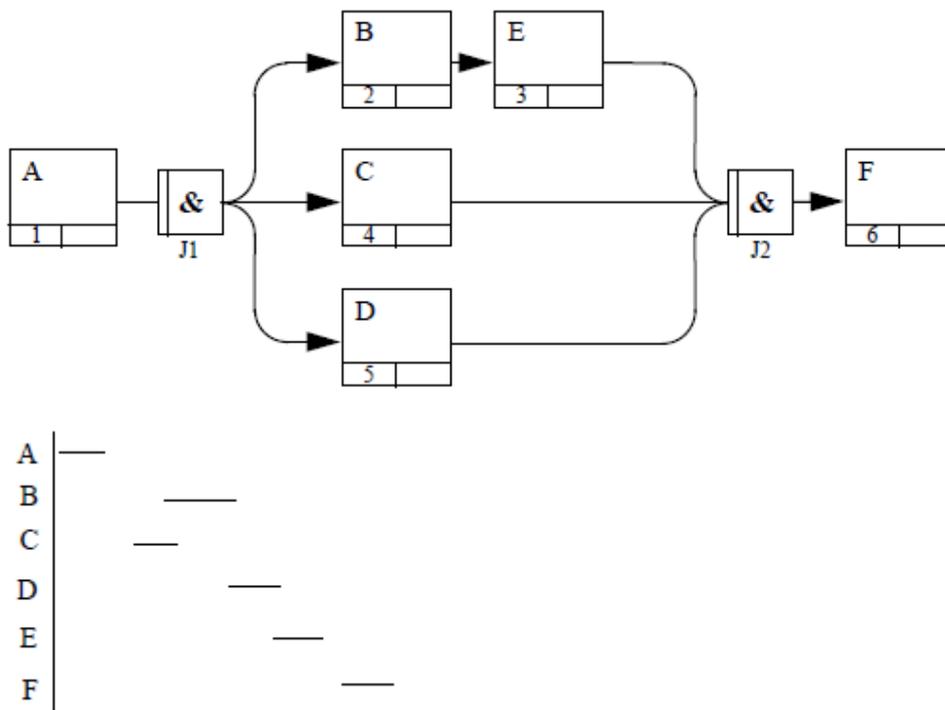


Рис. 2.12. Пример использования перекрестков

Референты расширяют границы понимания диаграммы и упрощают конструкцию описания. Референты используются для того, чтобы уточнить понимание процесса и добавить дополнительный смысл в систему.

Типы референтов приведены в таблице 2.2.

Таблица 2.2

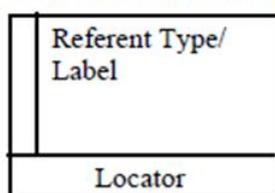
Типы референтов

Тип референта	Обозначение референта
UOB	Имя функционального элемента UOB (№ UOB)
SCENARIO	Название сценария (№ Scenario)
TS (Transition Schematic)	Название диаграммы перехода состояний (№ диаграммы перехода)
GO-TO	Имя функционального элемента UOB (№ UOB, № сценария или декомпозиции, в которой находится элемент)

Графические символы для двух основных стилей референтов отображены на рис. 2.13. Каждый тип референта может быть использован либо в схеме процесса, либо на объекте схемы. Чаще всего используется референт *Вызова-и-Продолжения*, который возникает в процессе исполнения блока и нужен до того, как блок, его вызвавший, завершится.

Референт *Вызов-и-Ожидание* должен начаться и завершиться до того, как блок, его вызвавший, завершится.

Call and Continue Referent



Call and Wait Referent

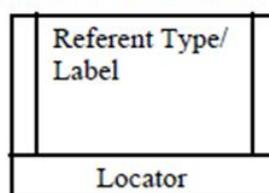


Рис. 2.13. Референты Вызова-и-Продолжения и Вызова-и-Ожидания

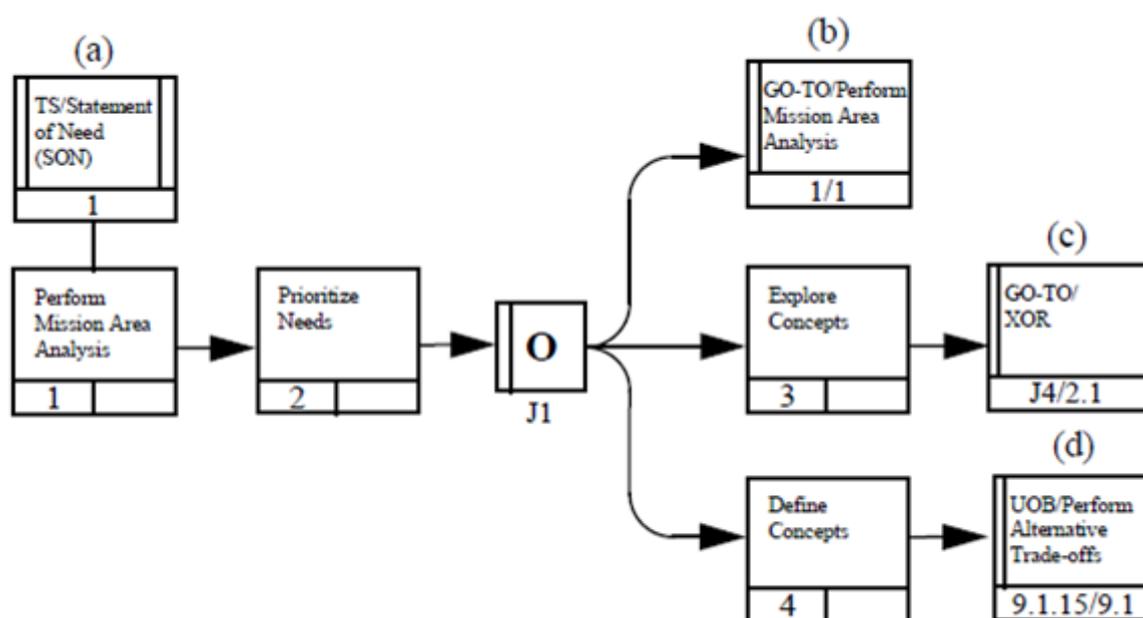


Рис. 2.14. Пример использования элемента «референт»

Элемент «примечание». Этот элемент используется для обеспечения дополнительной информации в процессе моделирования, присоединения к диаграммам иллюстраций, текста, комментариев и т.д. Они предоставляют возможность выразить идеи или концепции вместо использования относительных связей. Этот элемент может быть приложен к функциональному элементу, перекрестку, связи, объекту или референту и предназначен:

- для идентификации специфических объектов или отношений, связанных с функциональным элементом *UOB* связью или переходом;
- присоединения примеров, объектов, например экранных форм и т.п.;
- отображения специальных условий, уточнений соединения или ограничений, связанных с элементами диаграмм.

Декомпозиция описания процесса

Методология IDEF3 дает возможность представлять процесс в виде иерархически организованной совокупности диаграмм. Диаграммы состоят из нескольких элементов описания процесса IDEF3, причем каждый функциональный элемент UOB потенциально может быть детализирован на другой диаграмме. Такое разделение сложных комплексных процессов на их структурные части называется *декомпозицией*.

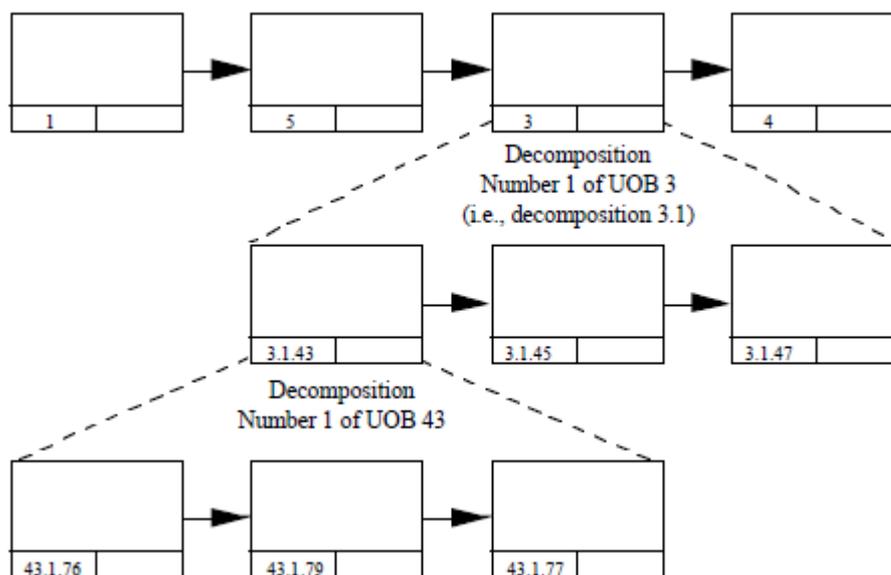


Рис. 2.15. Пример декомпозиции диаграммы IDEF3

2.5. Методология построения объектно-ориентированных систем IDEF4

IDEF4 - это объектно-ориентированный метод проектирования различных систем. Он был разработан для обеспечения перехода от предметной области и требований к объектно-ориентированным моделям, где отражается их структура, принципы взаимодействия. Объекты проектируются с достаточной детализацией, что дает возможность анализировать их исходную сущность. IDEF4 обеспечивает связь и дает возможность перейти от основного анализа объектов к их детальному рассмотрению и реализации. IDEF4 обеспечивает связь между результатами базового анализа предметной области и реализацией системы.

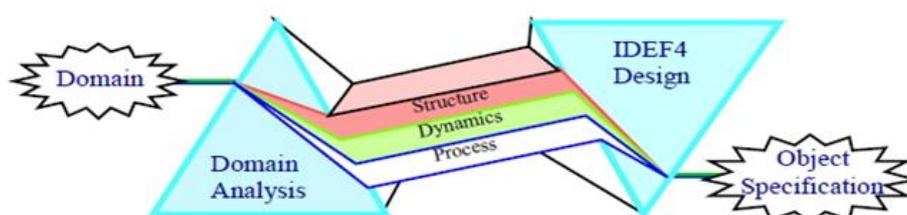


Рис. 2.16. Связь между результатами анализа предметной области и реализацией готовой системы

Философия объектно-ориентированного проектирования подчеркивает, что разделение аспектов проектирования на внешние и внутренние приводит к большому успеху, потому что это позволяет повторно использовать компоненты проектирования, обеспечивать параллельное проектирование, модульность конструкции (рис. 2.17).

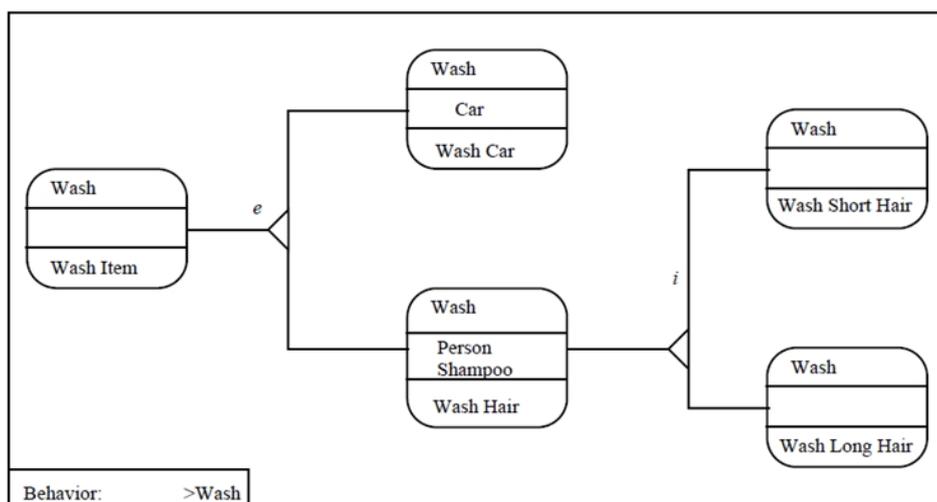


Рис. 2.17. Пример поведения системы в зависимости от результатов внутреннего и внешнего анализа

IDEF4 - метод многомерного подхода к объектно-ориентированному программному построению системы, в котором конструкция состоит из следующих элементов:

- уровень проектирования (на уровне системы, на уровне приложений, и базовый (нижний) уровень дизайна);
- определение статуса объекта дизайна (предметная область в приложении, на этапе обеспечения связи, в программном обеспечении описания предметной области);
- построение (дизайн) моделей взаимодействия (статические, динамические и модели поведения);
- расчетное обоснование моделей и уточнение ее конструктивных особенностей, начиная от общего к частному.

IDEF4 предусматривает дизайн моделей в трех отдельных слоях: (1) проектирование системы, (2) разработка приложений, и (3) базовый (нижний) уровень дизайна (рис. 2.18). Это трехслойная организация уменьшает сложность конструкции. При проектировании системы слой обеспечивает связь с другими системами. Слой разработки приложений отображает интерфейсы компонентов системы, которая конструируется. Эти компоненты включают в себя коммерческие приложения, ранее разработанные и реализованные, а также приложения, которые будут разработаны. Базовый (нижний) уровень дизайна представляет основные объекты системы.

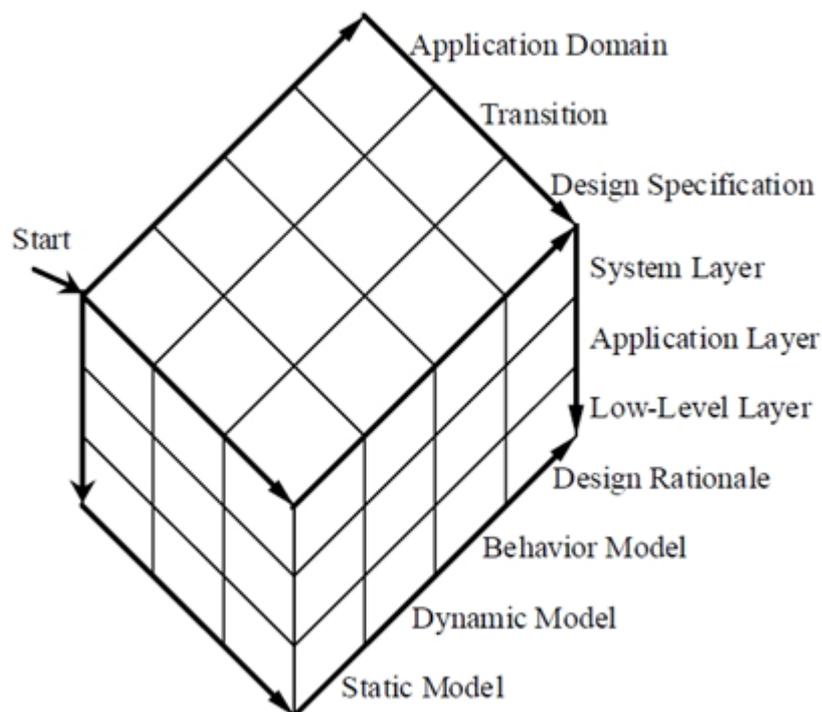


Рис. 2.18. Сущность IDEF4 модели

IDEF4 Статус явлений и объектов: любой созданный объект в IDEF4 может быть отмечен как основной (начальный), переходный или завершённый. Это позволяет проследить процесс создания системы от начала до завершения.

IDEF4 Дизайн Моделей: IDEF4 использует три дизайн-модели и обоснования компонентов в моделях. Статические Модели (SM) определяют неизменяемые во времени отношения между объектами (например, наследование). Динамические Модели (DM) определяют связи между объектами и переходы между состояниями объектов. Модели Поведения (БМ) определяют отношения между соответствующими объектами в процессе их функционирования. Расчетное обоснование компонентов моделей подразумевает представление системы сверху вниз, что позволяет дать представление об охвате этих трех моделей и документах, которые используются для обеспечения процесса основного дизайна системы.

В IDEF4 моделирование начинается с анализа требований и для начала моделируется предметная область объектов (application domain). При дальнейшем исследовании этих объектов они помечаются как «транзитные» и в конце как «финальные». Финал моделирования зависит от индивидуальных требований, заданного уровня детализации, размера объектов.

Статические модели, Динамические модели, Модели поведения и расчетное обоснование их компонент ложатся в основу базового (нижнего) уровня дизайна объектов. Вложенные слои могут быть построены в пределах каждого слоя, что позволяет уменьшить сложность систем.

IDEF4 – это многократная повторяющаяся (итерационная) процедура, состоящая из разделения, классификации/спецификации, сбора (соединения), моделирования и повторного перераспределения (перестановки) деятельности в рамках системы (рис. 2.19). Сначала система разделяется на объекты, каждый из которых либо сопоставляется с существующими объектами, либо для которых разрабатывается внешняя спецификация. Внешняя спецификация позволяет более корректно представить внутреннюю спецификацию.

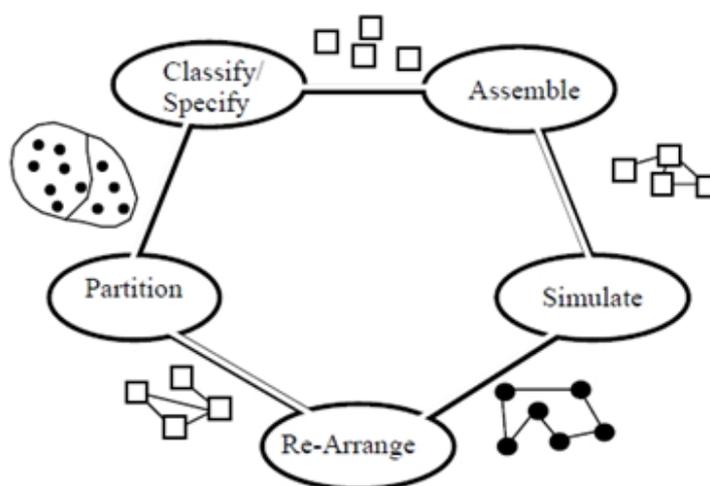


Рис. 2.19. Процесс разработки IDEF4 модели

После классификации/спецификации объекты соединяются в совокупность действий (бизнес-процессов) (т.е. статические, динамические и поведенческие модели подробно описывают различные аспекты взаимодействия между объектами). В процессе разработки моделей и систем важно моделировать сценарии и события для того, чтобы выявить недостатки конструкций. На основе выявленных недостатков проектировщик может переделать существующие модели и моделировать их до тех пор, пока они не станут корректными.

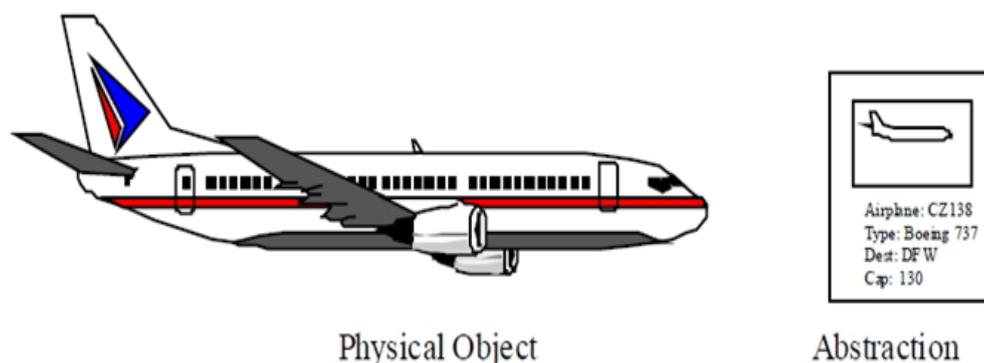


Рис. 2.20. Определение объекта «пассажирский самолет»

У объектов есть «поведение», которое описывает роль объекта в системе и «статус», которое описывает значение характеристики объекта.

Объекты идентифицируются в описании требований приложения. Они делятся на категории:

- физический объект, например, дерево, автомобиль, самолет, человек;
- объект-роль с особенностями поведения, которая устанавливает его значение, например, медсестра, пожарник или профессор;
- событие-объект, представляющий собой возникновение действий, например, встреча, поставка и др.;
- транзакции /взаимодействие – объекты, обеспечивающие взаимодействие других объектов, например, канал связи для пересылки сообщения;
- спецификация процедуры - объект, который представляет собой набор инструкций.

IDEF4 явления и объекты могут существовать в четырех различных абстракциях: приложения и предметные области объектов-предметов, объекты в натуральном выражении, спецификации объектов и программные объекты. В первых трех стадиях объекты отражаются так, как они существуют в природе. Программные объекты – это любые характеристики явлений. Например, “пассажирский самолет” становится объектом в предметной области, однако фактически этот объект определяет только «поведение» самолета по отношению к системе бронирования авиабилетов: количество мест, тип самолета, и полет назначения. Предметная область, где используется объект, является «сырьем» для дизайна объекта (рис. 2.20).

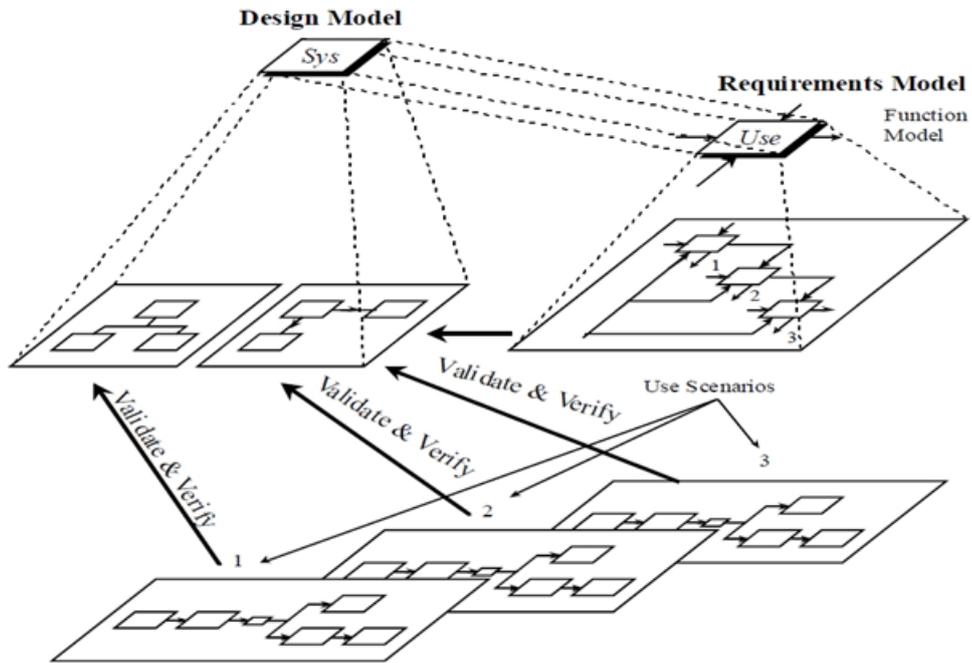


Рис. 2.21. Статическая, динамическая модели и модель требований для системного разделения

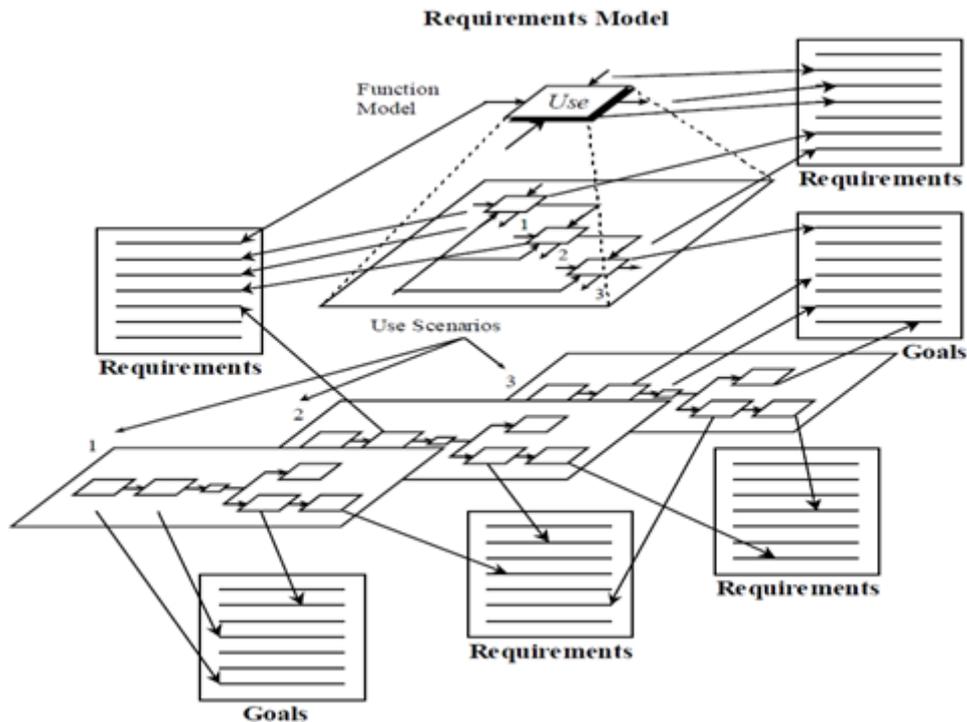


Рис. 2.22. Использование функциональных сценариев для определения «требований» и «целей»

2.6. Методология онтологического моделирования

Методология IDEF5 обеспечивает наглядное представление данных, полученных в результате обработки онтологических запросов в простой естественной графической форме.

Основной характерной чертой онтологического анализа является, в частности, разделение реального мира на составляющие и классы объектов и определение их онтологий, или же совокупности фундаментальных свойств, которые определяют их изменения и поведение. Таким образом, естественная наука представляет собой типичный пример онтологического исследования. Например, атомная физика классифицирует и изучает свойства наиболее фундаментальных объектов реального мира, таких как элементарные частицы, а биология, в свою очередь, описывает характерные свойства живых организмов, населяющих планету. Также существует большое количество сложных формаций или систем, созданных и поддерживаемых человеком, таких как производственные фабрики, военные базы, коммерческие предприятия и т.д. Эти формации представляют собой совокупность взаимосвязанных между собой объектов и процессов, в которых эти объекты тем или иным образом участвуют. Онтологическое исследование подобных сложных систем позволяет накопить ценную информацию об их работе, результаты анализа которой будут иметь решающее мнение при проведении процесса реорганизации существующих и построении новых систем.

Основные принципы онтологического анализа

Онтологический анализ обычно начинается с составления словаря терминов, который используется при обсуждении и исследовании характеристик объектов и процессов, составляющих рассматриваемую систему, а также создания системы точных определений этих терминов. Кроме того, документируются основные логические взаимосвязи между соответствующими введенным терминам понятиями. Результатом этого анализа является онтология системы, или же совокупность словаря терминов, точных их определений взаимосвязей между ними.

В любой системе существует две основные категории предметов восприятия, такие как сами объекты, составляющие систему (физические и интеллектуальные) и взаимосвязи между этими объектами, характеризующие

состояние системы. В терминах онтологии, понятие взаимосвязи, однозначно описывает или, другими словами, является точным дескриптором зависимости между объектами системы в реальном мире, а термины - являются, соответственно, точными дескрипторами самих реальных объектов.

При построении онтологии, в первую очередь происходит создание списка или базы данных дескрипторов и с помощью них, если их набор достаточен, создается модель системы. Таким образом, на начальном этапе должны быть выполнены следующие задачи:

- создание и документирования словаря терминов;
- описание правил и ограничений, согласно которым на базе введенной терминологии формируются достоверные утверждения, описывающие состояние системы;
- построение модели, которая на основе существующих утверждений, позволяет формировать необходимые дополнительные утверждения.

Процесс построения онтологии, согласно методологии IDEF5 состоит из пяти основных действий:

- 1) изучение и систематизирование начальных условий. Это действие устанавливает основные цели и контексты проекта разработки онтологии, а также распределяет роли между членами проекта;
- 2) сбор и накапливание данных. На этом этапе происходит сбор и накапливание необходимых начальных данных для построения онтологии;
- 3) анализ данных. Эта стадия заключается в анализе и группировке собранных данных и предназначена для облегчения построения терминологии;
- 4) начальное развитие онтологии. На этом этапе формируется предварительная онтология, на основе отобранных данных;
- 5) уточнение и утверждение онтологии – заключительная стадия процесса.

Язык описания онтологий в IDEF5

Для поддержания процесса построения онтологий в IDEF5 существуют специальные онтологические языки: схематический язык (Schematic Language-SL) и язык доработок и уточнений (Elaboration Language-EL). SL является наглядным графическим языком, специально предназначенным для изложения компетентными специалистами в рассматриваемой области сис-

темы основных данных в форме онтологической информации (рис. 2.23). Этот язык позволяет естественным образом представлять основную информацию в начальном развитии онтологии и дополнять существующие онтологии новыми данными. EL представляет собой структурированный текстовый язык, который позволяет детально характеризовать элементы онтологии.

Язык SL позволяет строить разнообразные типы диаграмм и схем в IDEF5. Основная цель всех этих диаграмм – наглядно и визуально представлять основную онтологическую информацию.

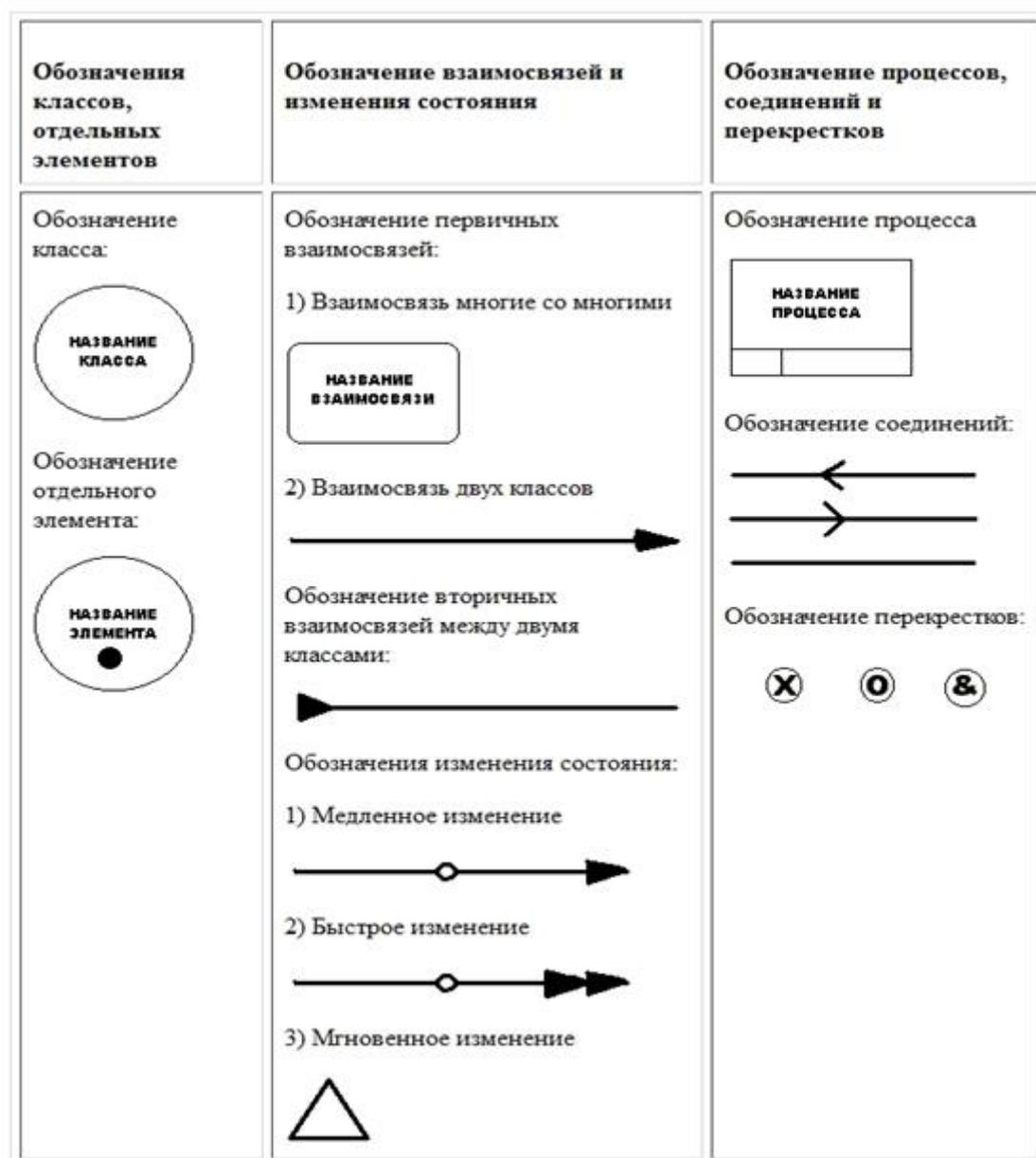


Рис. 2.23. Схематические графические изображения IDEF5

Виды схем и диаграмм IDEF5

Как правило, наиболее важные и заметные зависимости между объектами всегда являются преобладающими, когда конкретные люди высказывают свои знания и мнения, касающиеся той или иной системы. Подобные взаимосвязи явным образом описываются языками IDEF5. Всего существует четыре основных вида схем, которые наглядно используются для накопления информации об онтологии в достаточно прозрачной графической форме.

Диаграмма классификации обеспечивает механизм для логической систематизации знаний, накопленных при изучении системы. Существует два типа таких диаграмм: *диаграмма строгой классификации (Description Subsumption - DS)* и *диаграмма естественной или видовой классификации (Natural Kind Classification - NKC)*. Основное отличие диаграммы DS заключается в том, что определяющие свойства классов высшего и всех последующих уровней являются необходимым и достаточным признаком принадлежности объекта к тому или иному классу. На рис. 2.24 (а) приведен пример такой диаграммы. С помощью диаграмм DS, как правило, классифицируются логические объекты.

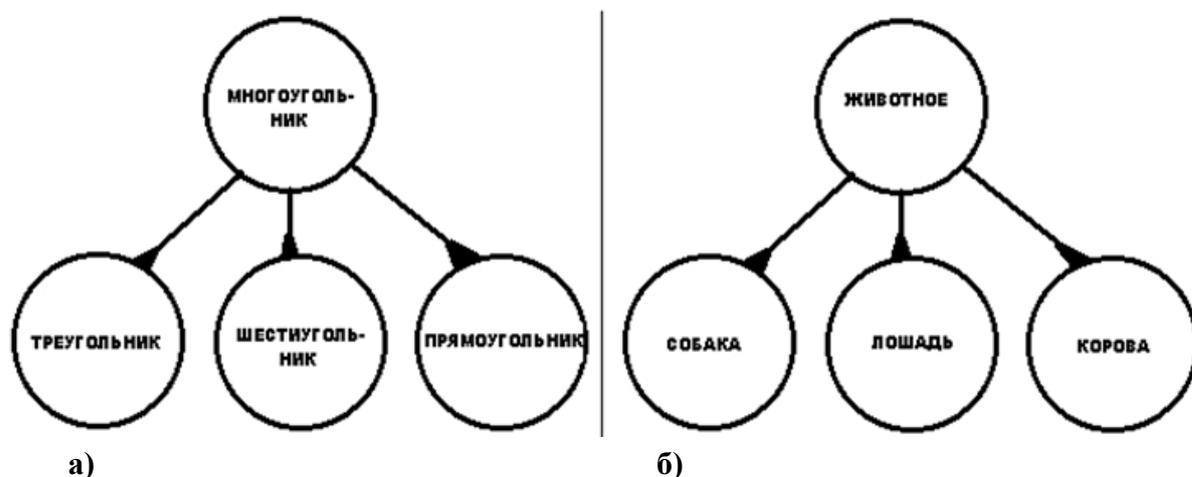


Рис. 2.24. Виды диаграмм IDEF5: диаграмма строгой классификации (а) и диаграмма естественной классификации (б)

Диаграммы естественной классификации или же *диаграммы NKC*, наоборот, не предполагают того, что свойства класса являются необходимым и достаточным признаком для принадлежности к ним тех или иных объектов. В этом виде диаграмм определение свойств класса является более общим. Пример такой диаграммы также приведен на рис. 2.24 (б).

Композиционная схема (Composition Schematics) являются механизмом графического представления состава классов онтологии и фактически пред-

ставляют собой инструменты онтологического исследования по принципу "Что из чего состоит". В частности, композиционные схемы позволяют наглядно отображать состав объектов, относящихся к тому или иному классу. На рис. 2.25 изображена композиционная схема шариковой ручки, относящейся к классу шариковых автоматических ручек.

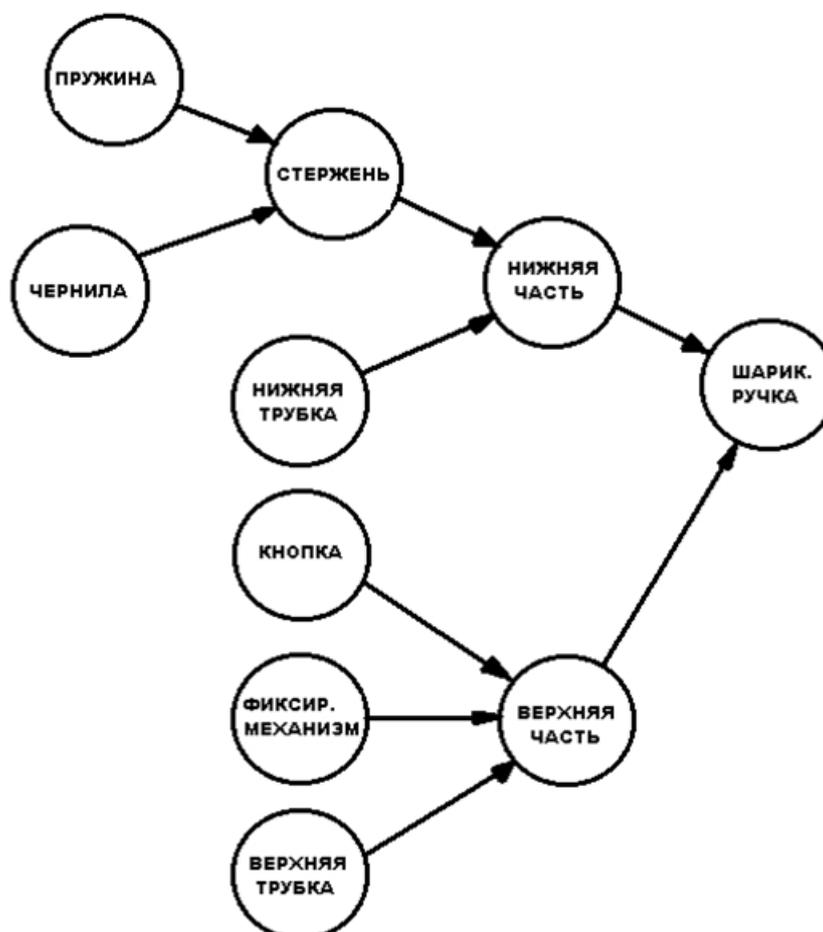


Рис. 2.25. Пример композиционной схемы

Схема взаимосвязей (Relation Schematics) позволяет разработчикам визуализировать и изучать взаимосвязи между различными классами объектов в системе. В некоторых случаях схемы взаимосвязей используются для отображения зависимостей между самими же классовыми взаимосвязями. Мотивацией для развития подобной возможности послужило то тривиальное правило, что все вновь разработанные концепции всегда базируются на уже существующих и изученных. Это тесно согласуется с теорией Новака и Гоуэна (Novak & Gowin, 1984), суть которой в том, что изучение любой системы часто происходит от частного к общему, то есть, происходит изыскание и исследование новой частной информации, влияющее на конечные характеристики более общей концепции, к которой эта информация имела прямое от-

ношение. Исходя из этой гипотезы, естественным образом изучения новой или плохо понимаемой взаимосвязи является соотнесение ее с достаточно изученной взаимосвязью, для исследования характеристик их сосуществования.

Диаграмма состояния объекта позволяет документировать тот или иной процесс с точки зрения изменения состояния объекта. В происходящих процессах могут произойти два типа изменения объекта: объект может поменять свое состояние или класс. Между этими двумя видами изменений по сути не существует принципиальной разницы: объекты, относящиеся к определенному классу в начальном состоянии, в течение процесса могут просто перейти к его дочернему или просто родственному классу. Например, полученная в процессе нагревания теплая вода, уже относится не к классу «вода», а к его дочернему классу «теплая вода». Однако при формальном описании процесса, во избежание путаницы, целесообразно разделять оба вида изменений, и для такого разделения используется обозначения следующего вида: "класс: состояние". Например, теплая вода будет описываться следующим образом: "вода: теплая", холодная - "вода: холодная" и так далее. Таким образом, диаграммы состояния в IDEF5 наглядно представляют изменения состояния или класса объекта в течение всего хода процесса. Пример такой диаграммы приведен на рис. 2.26.

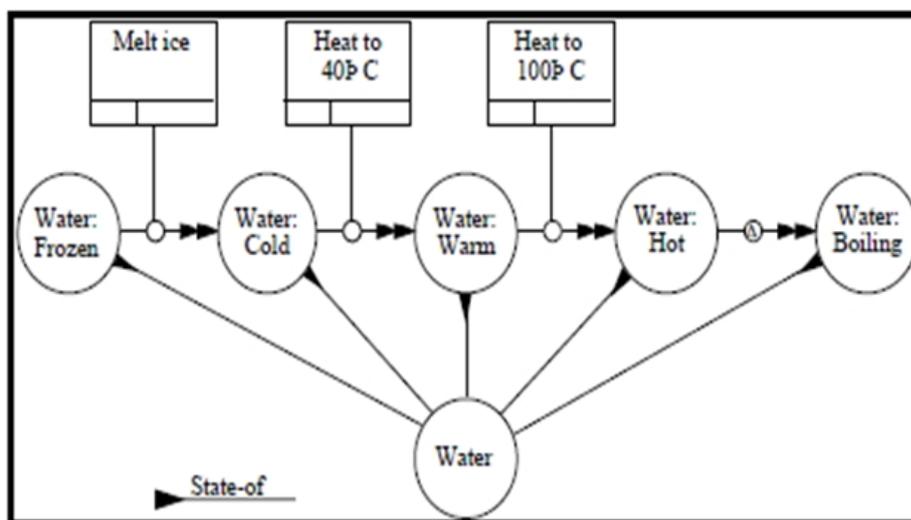


Рис. 2.26. Пример диаграммы состояния

Таким образом, строение и свойства любой системы могут быть эффективно исследованы и задокументированы при помощи следующих средств: словаря терминов, используемых при описании характеристик объектов и

процессов, имеющих отношение к рассматриваемой системе, точных и однозначных определений всех терминов этого словаря и классификации логических взаимосвязей между этими терминами. Набор этих средств является онтологией системы, а стандарт IDEF5 предоставляет структурированную методологию, с помощью которой можно наглядно и эффективно разрабатывать, поддерживать и изучать эту онтологию.

2.7. Обзор частично разработанных IDEF методов

2.7.1. IDEF6 – метод рационального представления процесса проектирования информационных систем

Метод позволяет обосновать необходимость проектируемых моделей, выявить причинно-следственные связи и отразить это в итоговой документации системы.

Развитие технологий привело к появлению товаров, которые приносят полезный эффект в течение десятилетий. Информационные системы также эволюционировали от отдельных приложений, ориентированных на системы с относительно коротким жизненным циклом, до крупномасштабных распределенных систем, которые должны обслуживать пользователей в течение длительного периода времени.

Обслуживание информационных систем, полезный эффект от использования которых ожидается в течение длительного периода времени, требует исследования и обоснования целесообразности их создания и эксплуатации.

Расчетное обоснование, как правило, существует в виде неструктурированных текстовых комментариев. При этом отсутствует структурированный метод организации и определения критерия целесообразности дальнейшего процесса проектирования ИС.

В отличие от методик проектирования, которые отвечают на вопрос КАКОЙ дизайн необходимо создать, новые методы дают ответ на вопросы ПОЧЕМУ именно такой дизайн или ПОЧЕМУ он не принял какую-то другую форму и при этом КАК окончательный дизайн и конфигурация были достигнуты.

Предназначение IDEF6 заключается в том, чтобы методически обосновать целесообразность проектирование информационных систем и выявить причинно-следственные связи.

Таким образом, методика IDEF6 предпринимает попытки выявления логики, лежащей в основе решения и конечного дизайна. Четкое установление целесообразного дизайна помогает избежать повторения прошлых ошибок, предоставляет прямые результаты последствий предлагаемых изменений в конструкции, заставляет яснее изложить цели и предположения и в результате помогает определить окончательные спецификации системы. Четкое выявление мотивов, почему дизайнер выбрал и принял конкретный проект и дизайн, стратегию реализации системы на уровне предприятия, информационных систем, является необходимым для поддержания жизненного цикла самой системы.

Обоснование принятого дизайна важно в тех случаях, когда исходные данные и условия не полностью определены из-за ограничений в конкретной ситуации.

В соответствии с целью обоснования дизайна ставятся следующие задачи:

- 1) обеспечить эволюционный процесс интеграции информационных систем на предприятии;
- 2) дать возможность использовать параллельные инженерные методы в развитии информационных систем;
- 3) поддержать наилучшую интеграцию в течение жизненного цикла системы и связанных с ней явлений;
- 4) облегчить реинжиниринг деловых процессов путем использования решений в бизнес-кейсах;
- 5) обеспечить эффективное отслеживание решений.

На рис. 2.27 приведен пример рационализации системы.

Рационализация Дизайна DS1 -> DS2

Замечания: невозможно правильно удовлетворить требования 5.3.1 для Сотрудника и Несотрудника. Нехарактерная связь между Персоной и Компанией в PC1.

Действия по рационализации: разделить Персоны на Сотрудников и Несотрудников. Пересмотреть кардинально ограничения от R1 до 1. Поменять

выявленное в Объекте Сотрудник/Компания в Структуре Диаграммы ЕС1 и в наследовании Диаграммы И1.

Рационализация Дизайна DS2 -> DS3

Замечания: Связь между Сотрудником и Компанией в ЕС1 невыполнима.

Действия по рационализации: Создать ссылающиеся атрибуты: Сотрудника на Компанию, Работодателя на Сотрудника. Также создать способ для обеспечения связи. Поменять указанное в ЕС2.

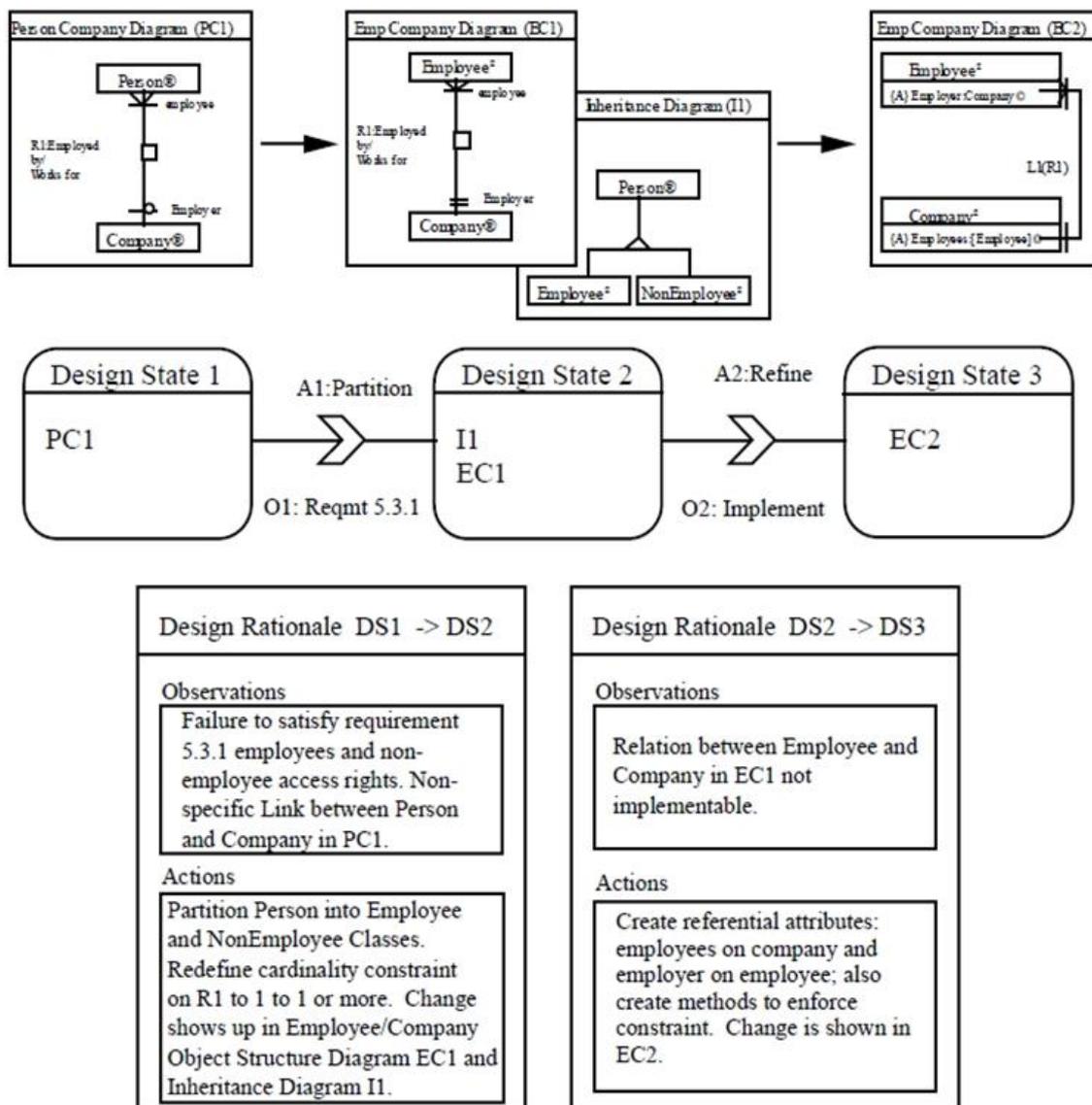


Рис. 2.27. Замечания/действия по рационализации дизайна системы

2.7.2. IDEF8 – стандарт описания интерфейсов взаимодействия оператора и системы

IDEF8 (Human-System Interaction Design) – стандарт описания интерфейсов взаимодействия оператора и системы (пользовательских интерфейсов). IDEF8 фокусирует внимание разработчиков интерфейса на программировании желаемого взаимного поведения интерфейса и пользователя на трех уровнях: выполняемой операции (что это за операция); сценарии взаимодействия, определяемом специфической ролью пользователя (по какому сценарию она должна выполняться тем или иным пользователем) и на деталях интерфейса (какие элементы управления предлагает интерфейс для выполнения операции).

Многое для методологии IDEF8 позаимствовано непосредственно из IDEF3, потому что IDEF8 нужен механизм для сбора и организации обработки информации на различных уровнях абстракции и детализации. Конструкции IDEF8 имеют предписывающий характер, а IDEF3 дает описательные представления.

IDEF8 стремится помочь пользователям обеспечить рациональное взаимодействие человека и системы (интерфейса), ориентироваться на пользователей, вовлечь пользователей к участию в проектной деятельности, сосредоточить усилия на проверки конструкций с помощью макетов и прототипов, а также оказать содействие созданию более продуктивной системы итераций через дизайн процесс.

Компонент, ответственный за сбор информации по взаимодействию человека и системы фиксирует, какие команды человек передает системе и как система на них реагирует. Дизайнеры должны учиться учитывать при проектировании специфику взаимодействия человека и технологии, доступной для конкретной системы.

В следующем списке перечислены ситуации, в которых результаты, полученные посредством использования инструментария IDEF8, будут создаваться и использоваться.

- 1) *Анализ.* IDEF8-модели используются как способ понимания и моделирования взаимодействия человека и системы. Взаимодействие с системой отражены в IDEF8-моделях. Моделирование существующих систем помогает выявить недостатки их проектирования или реализации.

- 2) *Проектирование.* IDEF8-модели используются для проектирования взаимодействия между пользователями и при этом системы могут разрабатываться на нескольких уровнях абстракции.
- 3) *Реализация.* IDEF8-модели могут использоваться, чтобы обеспечить дополнительными характеристиками (спецификациями) разработчиков.
- 4) *Документирование.* С помощью IDEF8-метода происходит документирование существующей системы или описание дизайна новой системы. В зависимости от потребностей проекта IDEF8-метод может использоваться для поддержки одной или более стадии процесса разработки. Пример использования метода IDEF8 приведен на рис. 2.28.

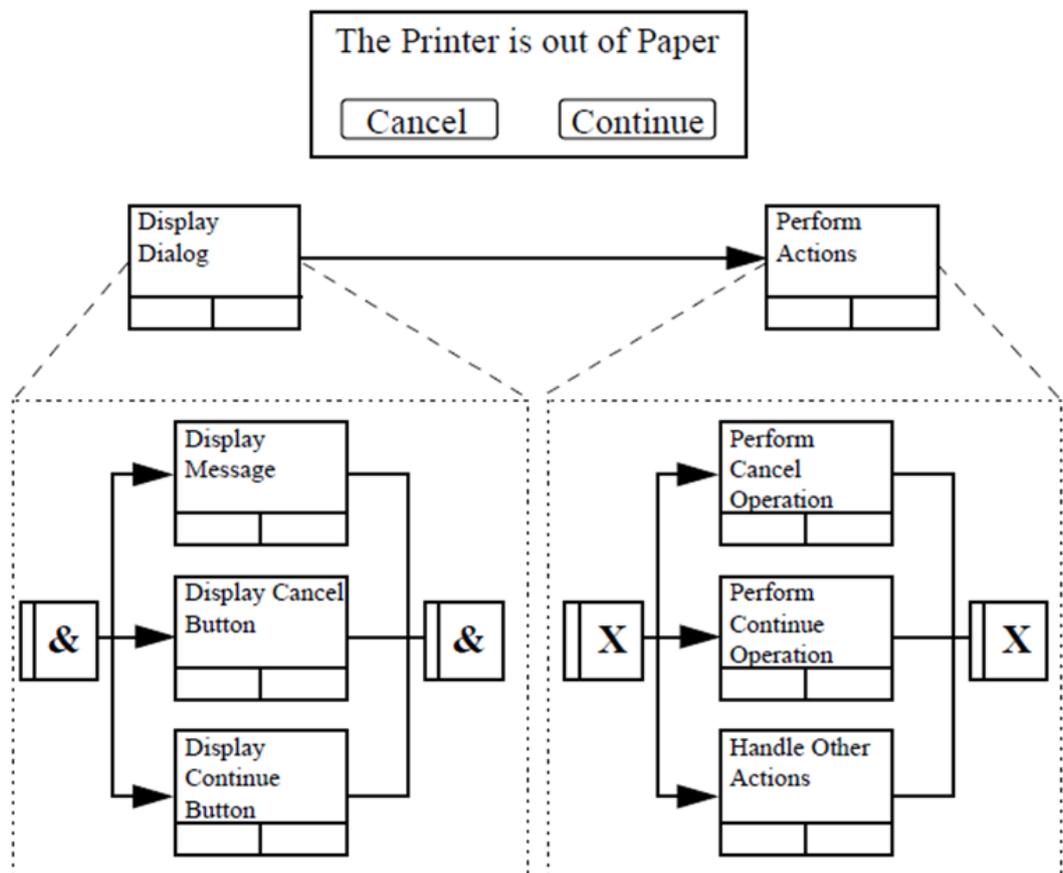


Рис. 2.28. Взаимодействие пользователя и системы при отсутствии бумаги в принтере

2.7.3. IDEF9 - метод анализа условий и ограничений

Модель IDEF9 (Business Constraint Discovery) предназначена для анализа имеющихся условий и ограничений (в том числе физических, юридических

или любых других) и их влияния на принимаемые решения в процессе реинжиниринга.

В IDEF9 методе рассматриваются и анализируются ограничения и условия, имеющиеся в бизнес-системе. К таким ограничениям следует относить принципы, правила, конвенции, процедуры, договора, соглашения, правила, а также социальные и физические законы. Эти элементы являются частью механизма при формировании отношений между людьми, информацией, материалами, машинами и системами. Если рассматривать предприятие как на систему, то ограничения формируют архитектуру и поведение этой системы.

Ограничения и условия фиксируются в документах и отчетности.

При этом при формировании условий и ограничений работы системы в ряде случаев возникают следующие проблемы следующего характера:

- затраты на обеспечение поддержания ограничений превышают стоимость самих ограничений;
- существующие ограничения выходят за рамки целей организации;
- ограничения вызывают непредвиденные и нежелательные последствия;
- агент или система (механизм), ответственные за поддержание ограничений не поддерживают корректно данное ограничение;
- то, что предполагалось как ограничение и условие, оказалось не способным поддержать и укрепить общий механизм.

Лица, ответственные за установление ограничений, должны определить их цели и масштабы при построении модели системы как можно раньше. Цель, как правило, устанавливается клиентом. Масштаб проекта определяется набором инструкций, которые формируют границы проекта. В проекте устанавливаются специфические зоны, для которых будут актуальны ограничения, а также зоны, на которых ограничения могут быть игнорированы.

Результаты использования методологии IDEF9 могут быть представлены в виде таблицы на рис. 2.29.

Ограничения могут быть в общем виде поделены на «дающие возможность», «стимулирующие» и «ограничивающие». Термин «ограничение» часто вызывает образы негативного влияния или жесткого контроля, но тем не менее ограничения играют важную стимулирующую роль в создании системы (например, размер допусков между сопрягаемыми деталями – необходимое условие для обеспечения правильной посадки изделия).

IDEF9 Project Summary Form	
Project Title: _____	
Project Leader: _____	
Purpose: _____	

Context: _____	

Major in-scope situations:	Major out-of-scope situations:
_____	_____
_____	_____
_____	_____
_____	_____
_____	_____
_____	_____

Рис. 2.29. Таблица для формирования ограничений и условий в методологии IDEF9

2.7.4. IDEF14 - метод проектирования вычислительных сетей

IDEF14 (Network Design Method) – метод проектирования вычислительных сетей, позволяющий устанавливать требования, определять сетевые компоненты, анализировать существующие сетевые конфигурации и формулировать желаемые характеристики сети.

IDEF14 может быть использован для моделирования существующих (AS-IS) компьютерных сетей или тех, которые должны быть (TO-BE). Это позволяет дизайнеру рассмотреть дизайн сети с точки зрения «ЧТО-ЕСЛИ» и оформить разумное объяснение. Основные цели IDEF14 метода исследования проектов были разработаны исходя из ощущаемых потребностей в качественных дизайнах сети, которые могут быть реализованы быстро и точно.

2.8. Диаграммы потоков данных (DFD)

Целью методики является построение модели рассматриваемой системы в виде диаграммы потоков данных (*Data Flow Diagram – DFD*), обеспечивающей правильное описание выходов (отклика системы в виде данных) при заданном воздействии на вход системы (подаче сигналов через внешние интерфейсы). Диаграммы потоков данных являются основным средством моделирования функциональных требований к проектируемой системе.

Для изображения DFD традиционно используются две различные нотации: Йодана (Yourdon) и Гейна-Карсона (Gane-Sarson), представленные на рис. 2.30.

Потоки данных являются механизмами, используемыми для моделирования передачи информации (или физических компонент) из одной части системы в другую. Потоки на диаграммах обычно изображаются именованными стрелками, ориентация которых указывает направление движения информации.

Назначение **процесса (работы)** состоит в продуцировании выходных потоков из входных в соответствии с действием, задаваемым именем процесса. Имя процесса должно содержать глагол в неопределенной форме с последующим дополнением (например, «вычислить максимальную высоту»). Кроме того, каждый процесс должен иметь уникальный номер для ссылок на него внутри диаграммы. Этот номер может использоваться совместно с номером диаграммы для получения уникального индекса процесса во всей модели.

Хранилище (накопитель) данных позволяет на определенных участках определять данные, которые будут сохраняться в памяти между процессами. Фактически хранилище представляет «срезы» потоков данных во времени. Информация, которую оно содержит, может использоваться в любое время после ее определения, при этом данные могут выбираться в любом порядке. Имя хранилища должно идентифицировать его содержимое и быть существительным.

Внешняя сущность представляет собой материальный объект вне контекста системы, являющейся источником или приемником системных данных. Ее имя должно содержать существительное, например, «склад товаров». Предполагается, что объекты, представленные такими узлами, не должны участвовать ни в какой обработке.

Компонента	Нотация Йодана	Нотация Гейна-Сарсона
поток данных	ИМЯ →	ИМЯ →
процесс	ИМЯ НОМЕР	НОМЕР ИМЯ
хранилище	ИМЯ	ИМЯ
внешняя сущность	ИМЯ	ИМЯ

Рис. 2.30. Основные символы диаграммы потоков данных DFD

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

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

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

Кроме основных элементов в состав DFD входят словари данных и миниспецификации.

Словари данных являются каталогами всех элементов данных, присутствующих в DFD, включая групповые и индивидуальные потоки данных, хранилища и процессы, а также все их атрибуты.

Миниспецификации обработки описывают DFD-процессы нижнего уровня. Фактически миниспецификации представляют собой алгоритмы описания задач, выполняемых процессами: множество всех миниспецификаций является полной спецификацией системы.

Для обеспечения декомпозиции данных и некоторых других сервисных возможностей к DFD добавляются следующие типы объектов:

- **групповой узел** предназначен для расщепления и объединения потоков. В некоторых случаях может отсутствовать, то есть фактически вырождаться в точку слияния/расщепления потоков на диаграмме;
- **узел-предок** позволяет увязывать входящие и выходящие потоки между детализируемым процессом и детализирующей DFD;
- **неиспользуемый узел** применяется в ситуации, когда декомпозиция данных производится в групповом узле, при этом требуются не все элементы входящего в узел потока;
- **узел изменения имени** позволяет неоднозначно именовать потоки, при этом их содержимое эквивалентно. Например, если при проектировании разных частей системы один и тот же фрагмент данных получил различные имена, то эквивалентность соответствующих потоков дан-

ных обеспечивается узлом изменения имени. При этом один из потоков данных является входным для данного узла, а другой – выходным;

- текст в свободном формате в любом месте диаграммы.

Возможный способ изображения этих узлов приведен на рис. 2.31.

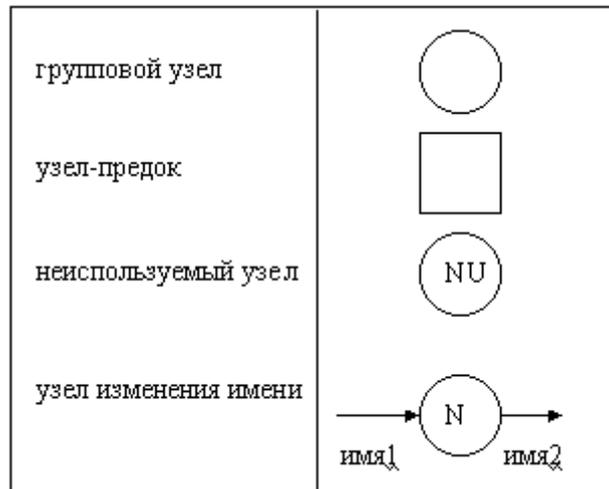


Рис 2.31. Расширения диаграммы потоков данных

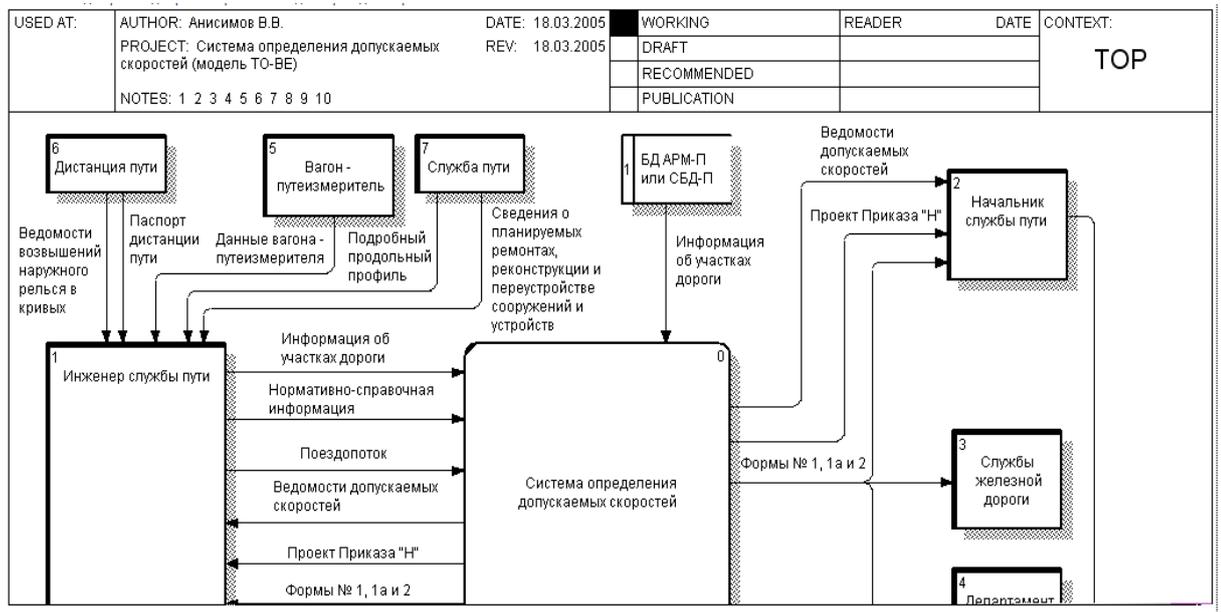


Рис. 2.32. Пример контекстной диаграммы DFD (в средстве моделирования VPwin)

Глава 3. Математические модели бизнес-процессов

3.1. Обзор математических моделей бизнес-процессов

Из знакомства с методологиями функционального моделирования бизнес-процессов следует, что они представляют собой формализованный подход к созданию функциональных моделей - структурных схем изучаемой системы. Схемы строятся по иерархическому принципу с необходимой степенью подробности и помогают разобраться в том, **что** происходит в изучаемой системе, какие функции в ней выполняются и в какие отношения вступают между собой и с окружающей средой ее функциональные блоки. Совокупность схем (IDEF0 - диаграмм) образует модель системы. Эта модель носит качественный, описательный, декларативный характер. Она принципиально не может ответить на вопросы о том, **как** протекают процессы в системе во времени и в пространстве, каковы их характеристики и в какой мере удовлетворяются (или не удовлетворяются) требования, предъявляемые к системе. Все эти вопросы с неизбежностью возникают после того, как достигнут нижний уровень декомпозиции, то есть обозначены функции нижнего уровня, с помощью которых и работает система.

В этом случае рекомендуется переходить к другим моделям – математическим, имитационным моделям, описывающим процессы в функциональных блоках IDEF (в частности, IDEF0 – модели). По терминологии, принятой в исследовании операций, IDEF0 - модели относятся к классу **концептуальных**. Именно концептуальные модели являются основой построения **математических моделей**. Пытаться «нагрузить» концептуальную модель количественными соотношениями не следует – это разные уровни абстракции.

Описание и количественная оценка преобразований требуют создания математических моделей, которые должны отображать (имитировать) физические, экономические, организационные, финансовые, логические и т.п. отношения между сущностями, входящими в IDEF0 – модель, разворачивающиеся во времени.

Исходя из общих соображений, связанных с возможными областями применения функционального моделирования и структурного анализа предприятий и организаций, можно указать несколько классов математических моделей, которые найдут применение в качестве средств описания процессов

и явлений, протекающих в IDEF0 - блоках. К их числу, в первую очередь, относятся:

- распределительные модели теории исследования операций (оптимальное распределение ресурсов);
- модели теории массового обслуживания (детерминированные и статистические);
- модели теории управления запасами;
- транспортные модели;
- динамические модели передачи сигналов (детерминированные и стохастические);
- регрессионные и корреляционные прогностические модели (в т.ч. модели, предсказывающие вероятность возникновения редких событий);
- некоторые модели теории игр.

Распределительные модели могут найти применение в тех случаях, когда требуется оптимальное распределение ресурсов, например, финансовых или трудовых, необходимых для выполнения некоторого подмножества операций IDEF0 - модели.

Модели теории массового обслуживания и управления запасами могут оказаться наиболее применимыми, поскольку многие процессы в организационно – экономических и производственно-технических системах - это процессы получения и обслуживания заявок на работы (услуги), а также процессы накопления, расходования, хранения и пополнения запасов, причем и те, и другие процессы необходимо вести с максимальной эффективностью.

Модели обслуживания позволяют оценивать производительность блоков, выполняющих те или иные операции обработки (преобразования) материальных и информационных объектов, определять реальную пропускную способность каналов, по которым передаются эти объекты, выявлять узкие места и резервы, оценивать зависимость производительности (пропускной способности) от надежности элементов, а также от расходования ресурсов (например, от текущих и капитальных затрат).

Транспортные модели позволяют не только оптимальным в каком-либо смысле образом планировать перевозки грузов, но и в более общем случае управлять передачей материальных или информационных объектов из пунктов их возникновения в пункты потребления или переработки.

Динамические модели передачи сигналов позволяют оценивать временные характеристики (запаздывания) передачи информации и помехозащищенность информационных каналов.

Прогностические модели позволяют решать задачи оптимального планирования с учетом тенденций развития изучаемой системы и ее компонентов.

Модели теории игр могут использоваться в качестве средств поддержки принятия решений при анализе структур, описываемых функциональными моделями.

3.2. Моделирование экономических систем с использованием марковских случайных процессов

3.2.1. Основные понятия марковских процессов

Функция $X(t)$ называется **случайной**, если ее значение при любом аргументе X является случайной величиной.

Случайная функция $X(t)$, аргументом которой является время, называется **случайным процессом**.

Марковские процессы являются частным видом случайных процессов. Особое место марковских процессов среди других классов случайных процессов обусловлено следующими обстоятельствами: для марковских процессов хорошо разработан математический аппарат, позволяющий решать многие практические задачи; с помощью марковских процессов можно описать (точно или приближенно) поведение достаточно сложных систем.

Случайный процесс, протекающий в какой-либо системе S , называется **марковским (или процессом без последствия)**, если он обладает следующим свойством: для любого момента времени t_0 вероятность любого состояния системы в будущем (при $t > t_0$) зависит только от ее состояния в настоящем (при $t = t_0$) и не зависит от того, когда и каким образом система S пришла в это состояние.

Классификация марковских случайных процессов производится в зависимости от непрерывности или дискретности множества значений функции $X(t)$ и параметра t .

Различают следующие основные виды марковских случайных процессов:

- с дискретными состояниями и дискретным временем (цепь Маркова);
- с непрерывными состояниями и дискретным временем (марковские последовательности);
- с дискретными состояниями и непрерывным временем (непрерывная цепь Маркова);
- с непрерывным состоянием и непрерывным временем.

Марковские процессы с дискретными состояниями удобно иллюстрировать с помощью *графа состояний* (рис. 3.1), где кружками обозначены состояния S_1, S_2, \dots , системы S , а стрелками – возможные переходы из состояния в состояние. На графе отмечаются только непосредственные переходы, а не переходы через другие состояния. Возможные задержки в прежнем состоянии изображают «петлей», то есть стрелкой, направленной из данного состояния в него же. Число состояний системы может быть как конечным, так и бесконечным (но счетным).

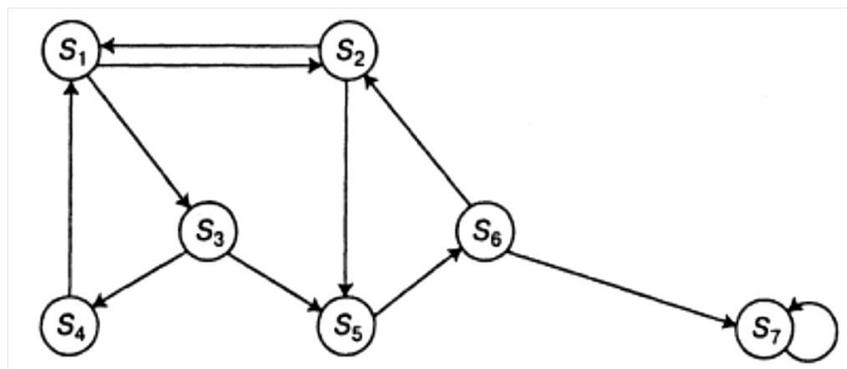


Рис. 3.1. Граф состояний системы S

3.2.2. Марковские цепи

Марковский случайный процесс с дискретными состояниями и дискретным временем называют *марковской цепью*. Для такого процесса моменты t_1, t_2, \dots , когда система S может менять свое состояние, рассматривают как последовательные шаги процесса, а в качестве аргумента, от которого зависит процесс, выступает не время t , а номер шага $1, 2, \dots, k, \dots$. Случайный процесс в этом случае характеризуется последовательностью состояний $S(0), S(1), S(2), S(k)$, где $S(0)$ – начальное состояние системы (перед первым шагом); $S(1)$ – состояние системы после первого шага; $S(k)$ – состояние системы после k -го шага.

Событие $\{S(k) = S_i\}$, состоящее в том, что сразу после k -го шага система находится в состоянии S_i ($i = 1, 2, \dots$), является случайным событием. По-

последовательность состояний $S(0), S(1), \dots, S(k)$ можно рассматривать как последовательность случайных событий. Такая случайная последовательность событий называется марковской цепью, если для каждого шага вероятность перехода из любого состояния S_i в любое S_j не зависит от того, когда и как система пришла в состояние S_i . Начальное состояние $S(0)$ может быть заданным заранее или случайным.

Вероятностями состояний цепи Маркова называются вероятности $P_j(k)$ того, что после k -го шага (и до $(k+1)$ -го) система S будет находиться в состоянии S_i ($i = 1, 2, \dots, n$). Очевидно, для любого k

$$\sum_{i=1}^n P_i(k) = 1$$

Начальным распределением вероятностей марковской цепи называется распределение вероятностей состояний в начале процесса $P_1(0), P_2(0), \dots, P_i(0), \dots, P_n(0)$.

В частном случае, если начальное состояние системы S в точности известно $S(0) = S_i$, то начальная вероятность $P_i(0) = 1$, а все остальные равны нулю.

Вероятностью перехода (переходной вероятностью) на k -м шаге из состояния S_i в состояние S_j называется условная вероятность того, что система S после k -го шага окажется в состоянии S_j при условии, что непосредственно перед этим (после $k - 1$ шага) она находилась в состоянии S_i .

Поскольку система может пребывать в одном из n состояний, то для каждого момента времени t необходимо задать n^2 вероятностей перехода P_{ij} , которые удобно представить в виде матрицы переходных вероятностей:

$$\|P_{ij}\| = \begin{bmatrix} P_{11} & P_{12} & \dots & P_{1n} \\ P_{21} & P_{22} & \dots & P_{2n} \\ \dots & \dots & \dots & \dots \\ P_{i1} & P_{i2} & \dots & P_{in} \\ \dots & \dots & \dots & \dots \\ P_{n1} & P_{n2} & \dots & P_{nn} \end{bmatrix},$$

где P_{ij} - вероятность перехода за один шаг из состояния S_i в состояние S_j ,
 P_{ij} — вероятность задержки системы в состоянии S_j .

Если переходные вероятности не зависят от номера шага (от времени), а зависят только от того, из какого состояния в какое осуществляется переход, то соответствующая цепь Маркова называется однородной.

Переходные вероятности однородной марковской цепи P_{ij} образуют квадратную матрицу размера $n \times n$, особенности которой заключаются в следующем:

1. каждая строка характеризует выбранное состояние системы, а ее элементы представляют собой вероятности всех возможных переходов за один шаг из выбранного (из i -го) состояния, в том числе и переход в самое себя;
2. элементы столбцов показывают вероятности всех возможных переходов системы за один шаг в заданное (j -е) состояние (иначе говоря, строка характеризует вероятность перехода системы из состояния, столбец – в состояние);
3. сумма вероятностей каждой строки равна единице, так как переходы образуют полную группу несовместных событий:

$$\sum_{j=1}^n P_{ij} = 1, i = \overline{1, n}$$

4. по главной диагонали матрицы переходных вероятностей стоят вероятности P_{ii} того, что система не выйдет из состояния S_i , а останется в нем. Если для однородной марковской цепи заданы начальное распределение вероятностей и матрица переходных вероятностей $\|P_{ij}\|$, то вероятности состояний системы $P_i(k) (i = \overline{1, n}; j = \overline{1, n})$ определяются по рекуррентной формуле:

$$P_i(k) = \sum_{j=1}^n P_j(k-1) * P_{ji} \quad (i = \overline{1, n}; j = \overline{1, n})$$

Пример 3.1. Рассмотрим процесс функционирования системы - автомобиль. Пусть автомобиль (система) в течение одной смены (суток) может находиться в одном из двух состояний: исправном (S_1) и неисправном (S_2). Граф состояний системы представлен на рис. 3.2.

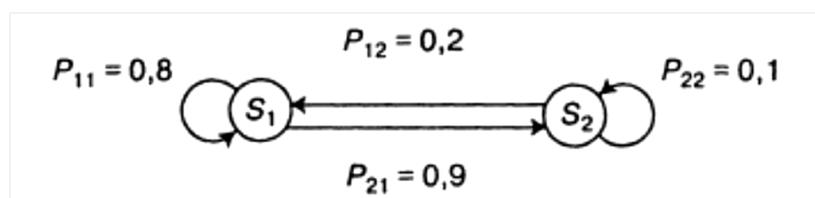


Рис. 3.2. Граф состояний автомобиля

В результате проведения массовых наблюдений за работой автомобиля составлена следующая матрица вероятностей перехода:

$$\|P_{ij}\| = \begin{bmatrix} 0,8 & 0,2 \\ 0,9 & 0,1 \end{bmatrix}$$

где $P_{11} = 0,8$ – вероятность того, что автомобиль останется в исправном состоянии;

$P_{12} = 0,2$ – вероятность перехода автомобиля из состояния «исправен» в состояние «неисправен»;

$P_{21} = 0,9$ – вероятность перехода автомобиля из состояния «неисправен» в состояние «исправен»;

$P_{22} = 0,1$ – вероятность того, что автомобиль останется в состоянии «неисправен».

Вектор начальных вероятностей состояний автомобиля задан $P(0) = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$,

$$P_1(0) = 0 \text{ и } P_2(0) = 1.$$

Требуется определить вероятности состояний автомобиля через трое суток.

Решение

Используя матрицу переходных вероятностей, определим вероятности состояний $P_i(k)$ после первого шага (после первых суток):

$$P_1(1) = P_1(0) * P_{11} + P_2(0) * P_{21} = 0 * 0,8 + 1 * 0,9 = 0,9$$

$$P_2(1) = P_1(0) * P_{12} + P_2(0) * P_{22} = 0 * 0,2 + 1 * 0,1 = 0,1.$$

Вероятности состояний после второго шага (после вторых суток) таковы:

$$P_1(2) = P_1(1) * P_{11} + P_2(1) * P_{21} = 0,9 * 0,8 + 0,1 * 0,9 = 0,81;$$

$$P_2(2) = P_1(1) * P_{12} + P_2(1) * P_{22} = 0,9 * 0,2 + 0,1 * 0,1 = 0,19.$$

Вероятности состояний после третьего шага (после третьих суток) равны:

$$P_1(3) = P_1(2) * P_{11} + P_2(2) * P_{21} = 0,81 * 0,8 + 0,19 * 0,9 = 0,819;$$

$$P_2(3) = P_1(2) * P_{12} + P_2(2) * P_{22} = 0,81 * 0,2 + 0,19 * 0,1 = 0,181.$$

Таким образом, после третьих суток автомобиль будет находиться в исправном состоянии с вероятностью 0,819 и в состоянии «неисправен» с вероятностью 0,181.

3.2.3. Непрерывные цепи Маркова

Марковский случайный процесс с дискретными состояниями и непрерывным временем называется **непрерывной цепью Маркова** при условии, что переход системы из состояния в состояние происходит не в фиксированные, а в случайные моменты времени.

В экономике часто встречаются ситуации, которые указать заранее невозможно (например, любая деталь или агрегат автомобиля могут выйти из строя в любой, непредсказуемый заранее момент времени). Для описания таких систем в отдельных случаях можно использовать математический аппарат непрерывной цепи Маркова.

Пусть система характеризуется n состояниями $S_0, S_1, S_2, \dots, S_n$, а переход из состояния в состояние может осуществляться в любой момент времени. Обозначим через $P_i(t)$ вероятность того, что в момент времени t система S будет находиться в состоянии S_i ($i = 0, 1, \dots, n$). Требуется определить для любого t вероятности состояний $P_0(t), P_1(t), \dots, P_n(t)$. Очевидно, что $\sum_{i=0}^n P_i(t) = 1$.

Для процесса с непрерывным временем вместо переходных вероятностей P_{ij} рассматриваются плотности вероятностей перехода λ_{ij} , представляющие собой предел отношения вероятности перехода системы за время Δt из состояния S_i в состояние S_j к длине промежутка Δt :

$$\lambda_{ij}(t) = \lim_{\Delta t \rightarrow 0} \frac{P_{ij}(t; \Delta t)}{\Delta t},$$

где $P_{ij}(t, \Delta t)$ - вероятность того, что система, пребывавшая в момент t в состоянии S_i за время Δt перейдет из него в состояние S_j (при этом всегда $i \neq j$).

Если $\lambda_{ij} = \text{const}$ то процесс называется **однородным**, если плотность вероятности зависит от времени $\lambda_{ij} = \lambda_{ij}(t)$, то процесс - **неоднородный**. При рассмотрении непрерывных марковских процессов принято представлять переходы системы S из состояния в состояние как происходящие под влиянием некоторых потоков событий. **Потоком событий** называется последователь-

ность однородных событий, следующих одно за другим через случайные интервалы времени. Плотность вероятности перехода интерпретируется как интенсивность λ_{ij} соответствующих потоков событий. Если все эти потоки пуассоновские, то процесс, протекающий в системе S , будет марковским.

При изучении марковских случайных процессов с дискретными состояниями и непрерывным временем в графе состояний над стрелками, ведущими из состояния S_i в S_j , проставляют соответствующие интенсивности λ_{ij} . Такой граф состояний называют **размеченным**.

Пусть система S имеет конечное число состояний S_0, S_1, \dots, S_n . Случайный процесс, протекающий в этой системе, описывается вероятностями состояний $P_0(t), P_1(t), \dots, P_n(t)$, где $P_i(t)$ – вероятность того, что система S в момент t находится в состоянии S_i . Для любого t

$$\sum_{i=0}^n P_i(t) = 1$$

Вероятности состояний $P_i(t)$ находят путем решения системы дифференциальных уравнений (уравнений Колмогорова), имеющих вид

$$\frac{dP_i(t)}{dt} = \sum_{j=1}^n \alpha_{ji} P_j(t) - P_i(t) \sum_{j=1}^n \alpha_{ij} \quad (3.1)$$

где $i = 0, 1, \dots, n$.

Величина $\lambda_{ij} P_i(t)$ называется потоком вероятности перехода из состояния S_i в S_j , причем интенсивность потоков λ_{ij} может зависеть от времени или быть постоянной.

Уравнения (3.1) составляют по размеченному графу состояний системы, пользуясь следующим мнемоническим правилом: производная вероятности каждого состояния равна сумме всех потоков вероятности, идущих из других состояний в данное состояние, минус сумма всех потоков вероятности, идущих из данного состояния в другие.

Чтобы решить систему дифференциальных уравнения (3.1), нужно задать начальное распределения вероятностей $P_0(0), P_1(0), \dots, P_i(0), \dots, P_n(0)$. Для решения применяют численные методы.

Финальные вероятности состояний

Если процесс, протекающий в системе, длится достаточно долго, то имеет смысл говорить о предельном поведении вероятностей $P_i(t)$ при $t \rightarrow \infty$.

В некоторых случаях существуют финальные (предельные) вероятности состояний:

$$P_i = \lim_{t \rightarrow \infty} P_i(t)$$

где $i = 0, 1, \dots, n$,

не зависящие от того, в каком состоянии система S находилась в начальный момент. Говорят, что в системе S устанавливается предельный стационарный режим, в ходе которого она переходит из состояния в состояние, но вероятности состояний P_i , уже не меняются. Система, для которой существуют финальные вероятности, называется *эргодической*, а соответствующий случайный процесс – *эргодическим*.

Финальные вероятности состояний (если они существуют) могут быть получены путем решения системы линейных алгебраических уравнений, которые получаются из дифференциальных уравнений Колмогорова, если приравнять производные к нулю, а вероятностные функции состояний $P_1(t), \dots, P_n(t)$ в правых частях уравнений (3.1) заменить соответственно на неизвестные финальные вероятности P_1, \dots, P_n .

Таким образом, для системы S с n состояниями получается система n линейных однородных алгебраических уравнений с n неизвестными P_0, P_1, \dots, P_n , которые можно найти с точностью до произвольного множителя. Для нахождения точного значения P_0, P_1, \dots, P_n к уравнениям добавляют нормирующее условие $P_0 + P_1 + \dots + P_n = 1$, пользуясь которым можно выразить любую из вероятностей P_i через другие и отбросить одно из уравнений.

Пример 3.2. Имеется размеченный граф состояний системы S (рис. 3.3). Необходимо составить систему дифференциальных уравнений Колмогорова и записать начальные условия для решения этой системы, если известно, что в начальный момент система находилась в состоянии S_1 .

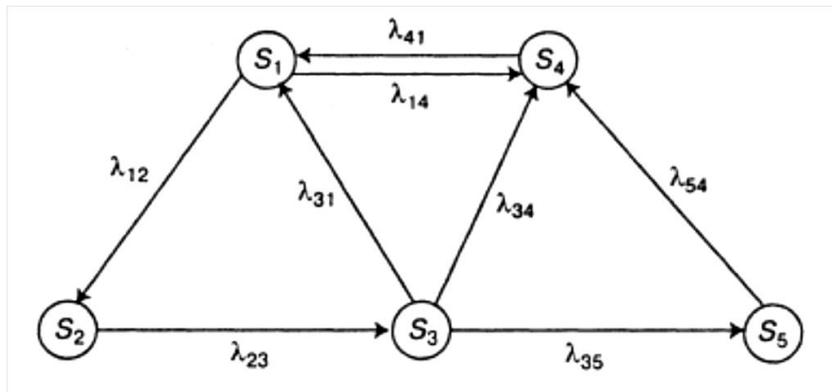


Рис. 3.3. Граф состояний системы

Решение

Согласно приведенному мнемоническому правилу, система дифференциальных уравнений Колмогорова имеет вид

$$\begin{cases} \frac{dP_1}{dt} = \alpha_{31} * P_3 + \alpha_{41} * P_4 - \alpha_{12} * P_1 - \alpha_{14} * P_1 \\ \frac{dP_2}{dt} = \alpha_{12} * P_1 - \alpha_{23} * P_2 \\ \frac{dP_3}{dt} = \alpha_{23} * P_2 - (\alpha_{31} * P_3 + \alpha_{34} * P_3 + \alpha_{35} * P_3) \\ \frac{dP_4}{dt} = \alpha_{41} * P_1 + \alpha_{34} * P_3 + \alpha_{54} * P_5 - \alpha_{41} * P_4 \\ \frac{dP_5}{dt} = \alpha_{35} * P_3 - \alpha_{54} * P_5 \end{cases} \quad (3.2)$$

Начальные условия при $t = 0$:

$$P_1=1; P_2=P_3=P_4=P_5=0$$

Рассмотрим, что произойдет с системой S описываемой дифференциальными уравнениями Колмогорова, при $t \rightarrow \infty$. Известно, что в случае сообщающихся состояний функции $P_1(t)$, $P_2(t)$, ... $P_n(t)$ стремятся к предельным (финальным) вероятностям состояний системы S . Финальные вероятности не зависят от времени. Поэтому в системе дифференциальных уравнений Колмогорова все левые части уравнений (производные) принимают равными нулю. При этом система дифференциальных уравнений превратится в систему линейных алгебраических уравнений.

Для нашего примера система (3.2) будет иметь вид

$$\begin{cases} 0 = \alpha_{31} * P_3 + \alpha_{41} * P_4 - \alpha_{12} * P_1 - \alpha_{14} * P_1 \\ 0 = \alpha_{12} * P_1 - \alpha_{23} * P_2 \\ 0 = \alpha_{23} * P_2 - (\alpha_{31} * P_3 + \alpha_{34} * P_3 + \alpha_{35} * P_3) \\ 0 = \alpha_{41} * P_1 + \alpha_{34} * P_3 + \alpha_{54} * P_5 - \alpha_{41} * P_4 \\ 0 = \alpha_{35} * P_3 - \alpha_{54} * P_5 \end{cases} \quad (3.3)$$

Решая ее, с учетом условия $P_1 + P_2 + P_3 + P_4 + P_5 = 1$ получим все предельные вероятности. Эти вероятности представляют собой не что иное, как среднее относительное время пребывания системы в данном состоянии.

Необходимые и достаточные условия существования финальных вероятностей

Для существования финальных вероятностей одного условия $\lambda_{ij} = const$ недостаточно, требуется выполнение еще некоторых условий, проверить которые можно по графу состояний, выделив в нем так называемые существенные и несущественные состояния.

Состояние S_i называется **существенным**, если нет другого состояния S_j , т. е. такого, что, перейдя однажды каким-то способом из S_i в S_j , система уже не может вернуться в S_i .

Все состояния, не обладающие таким свойством, называются **несущественными**.

Рассмотрим пример, представленный на рис. 3.4.

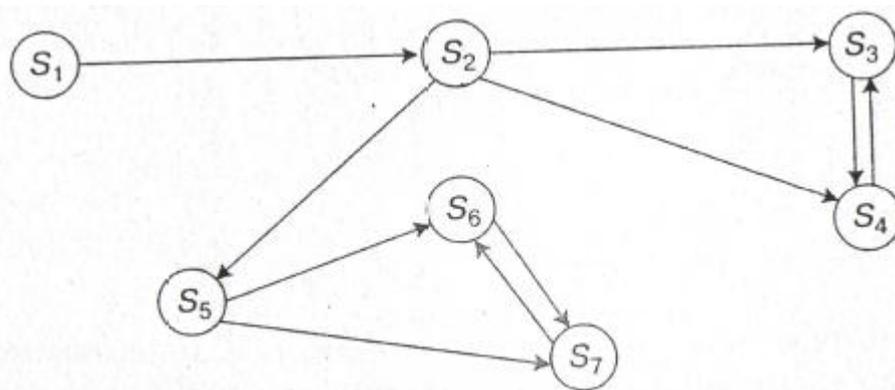


Рис. 3.4. Граф состояний системы

Состояния S_1, S_2 и S_5 - несущественные, так как из S_1 , можно уйти, например, в состояние S_2 и не вернуться, а из состояния S_2 в состояние S_3 или S_4 и не вернуться аналогично из состояния S_5 - в состояние S_6 и S_7 . Состояния S_3, S_4, S_6 и S_7 - существенные состояния.

Теорема. При конечном числе состояний для существования финальных вероятностей необходимо и достаточно, чтобы из каждого существенного состояния можно было (за какое-то число шагов) перейти в каждое другое существенное состояние.

Граф из примера рис. 3.4 этому условию не удовлетворяет, так как из существенного состояния S_4 нельзя перейти в существенное состояние S_7 .

При исследовании непрерывных марковских цепей, как было уже отмечено, часто бывает удобно представить переход системы из состояния в состояние как воздействие каких-то потоков событий (поток заявок на обслуживание, поток автомобилей, поток документов и т. п.). Различают следующие

щие основные свойства, которыми могут обладать случайные потоки событий:

- стационарность;
- ординарность;
- отсутствие последействия.

Стационарность. Свойство стационарности проявляется в том, что вероятность попадания того или иного числа событий на участок времени τ зависит только от длины участка и не зависит от расположения на оси ot . Другими словами, стационарность означает неизменность вероятностного режима потока событий во времени. Поток, обладающий свойством стационарности, называют стационарным. Для стационарного потока среднее число событий, воздействующих на систему в течение единицы времени, остается постоянным. Реальные потоки событий в экономике предприятия являются в действительности стационарными лишь на ограниченных участках времени.

Ординарность. Свойство ординарности потока присутствует, если вероятность попадания на элементарный участок времени двух и более событий пренебрежимо мала по сравнению с длиной этого участка. Свойство ординарности означает, что за малый промежуток времени практически невозможно появление более одного события. Поток, обладающий свойством ординарности, называют ординарным. Реальные потоки событий в различных экономических системах либо являются ординарными, либо могут быть достаточно просто приведены к ординарным.

Отсутствие последействия. Данное свойство потока состоит в том, что для любых непересекающихся участков времени количество событий, попадающих на один из них, не зависит оттого, сколько событий попало на другие участки времени. Поток, обладающий свойством отсутствия последействия, называют потоком без последействия. Поток событий, одновременно обладающий свойствами стационарности, ординарности и отсутствия последействия, называется *простейшим потоком событий*.

Для простейшего потока интенсивность $\alpha = const$. Если поток событий не имеет последействия, ординарен, но не стационарен, то его называют *нестационарным пуассоновским потоком*, а его интенсивность зависит от времени, т. е. $\alpha = \alpha(t)$.

В пуассоновском потоке событий (стационарном и нестационарном) число событий потока, попадающих на любой участок, распределено по *закону Пуассона*:

$$P_m = \frac{a^m}{m!} * e^{-a}, \quad m = 0, 1, \dots$$

где P_m – вероятность попадания на участок m событий;
 a – среднее число событий, приходящееся на участок.

Для простейшего потока $a = \alpha * \tau$, а для нестационарного пуассоновского потока

$$a = \int_{t_0}^{t_0 + \tau} \alpha(t) dt$$

где τ — длина участка времени;

t_0 — начало участка τ .

Отметим еще одно важное свойство простейшего потока событий. Промежуток времени t между соседними событиями распределен по показательному закону, а его среднее значение T и среднее квадратическое отклонение σ равны, то есть

$$T = \sigma = 1/\alpha$$

где α — интенсивность потока.

Для нестационарного пуассоновского потока закон распределения промежутка t уже не является показательным, так как зависит от положения на оси $0t$ и вида зависимости $\alpha(t)$. Однако для некоторых задач при сравнительно небольших изменениях $\alpha(t)$ его можно приближенно считать показательным с интенсивностью α , равной среднему значению $\alpha(t)$.

Таким образом, для исследуемой системы S с дискретными состояниями и непрерывным временем переходы из состояния в состояние происходят под действием пуассоновских потоков событий с определенной интенсивностью α_{ij} .

Рассмотрим еще одну типичную схему непрерывных марковских цепей – так называемую схему гибели и размножения, часто встречающуюся в разнообразных практических задачах.

Марковский процесс с дискретными состояниями $S_0, S_1, S_2, \dots, S_n$ называется процессом гибели и размножения, если все состояния можно вытянуть в одну цепочку, в которой каждое из средних состояний (S_1, S_2, \dots, S_{n-1}) может переходить только в соседние состояния, которые, в свою очередь,

переходят обратно, а крайние состояния (S_0 и S_n) переходят только в соседние состояния (рис. 3.5).

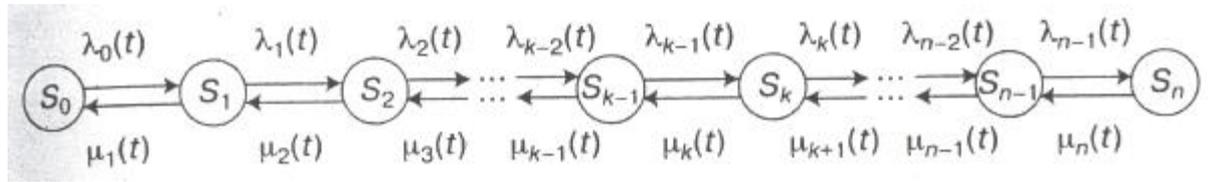


Рис. 3.5. Граф состояний для процесса гибели и размножения

Название взято из биологических задач, где состояние популяции S_k означает наличие в ней k единиц особей.

Переход вправо связан с размножением единиц, а влево — с их гибелью.

$\alpha_0(t), \alpha_1(t), \alpha_2(t), \dots, \alpha_{n-1}(t)$ — интенсивности размножения,

$\mu_1(t), \mu_2(t), \dots, \mu_n(t)$ — интенсивности гибели.

У α и μ индекс того состояния, из которого стрелка выходит.

Марковским процессом гибели и размножения с непрерывным временем называется такой случайный процесс, который может принимать только целые неотрицательные значения. Изменения этого процесса могут происходить в любой момент времени, т. е. в любой момент времени он может либо увеличиться на единицу, либо уменьшиться на единицу, либо остаться неизменным.

В практике встречаются процессы чистого размножения и чистой гибели. Процессом чистого размножения называется такой процесс гибели и размножения, у которого интенсивности всех потоков гибели равны нулю; аналогично процессом чистой «гибели» называется такой процесс гибели и размножения, у которого интенсивности всех потоков размножения равны нулю.

Пример 3.3. Рассмотрим эксплуатацию моделей автомобилей одной марки в крупной транспортной фирме (на предприятии). Интенсивность поступления автомобилей на предприятие равна $\alpha(t)$. Каждый поступивший на предприятие автомобиль списывается через случайное время T_c . Срок службы автомобиля T_c распределен по показательному закону с параметром μ . Процесс эксплуатации автомобилей является случайным процессом. $A(t)$ — число автомобилей данной марки, находящихся в эксплуатации в момент t .

Система уравнений Колмогорова для размеченного графа (рис. 3.7) имеет вид

$$\begin{aligned} dP_0(t)dt &= \mu * P_1(t) - \alpha(t) * P_0(t) \dots\dots\dots dP_i(t)dt = \alpha t * P_{i-1}(t) + i + 1 \\ &* \mu * P_{i+1}(t) - \alpha t + i * \mu * P_i(t), \\ i &= 1, n-1 \dots\dots\dots dP_n(t)dt = \alpha t * P_{n-1}(t) - n * \mu * P_n(t) \end{aligned} \quad (3.5)$$

Эту систему надо решать при начальных условиях, рассмотренных выше. Решения систем уравнений (3.4) и (3.5) являются одномерными законами распределения $P_i(t)$. Отыскание решений систем (3.4) и (3.5) в общем виде при произвольном виде функции $\alpha(t)$ представляет значительные трудности и не имеет практических приложений.

При постоянных интенсивностях потоков гибели и размножения и конечном числе состояний будет существовать стационарный режим. Система S с конечным числом состояний ($n + 1$), в которой протекает процесс гибели и размножения с постоянными интенсивностями потоков гибели и размножения, является простейшей эргодической системой. Размеченный граф состояний для такой системы представлен на рис. 3.8.

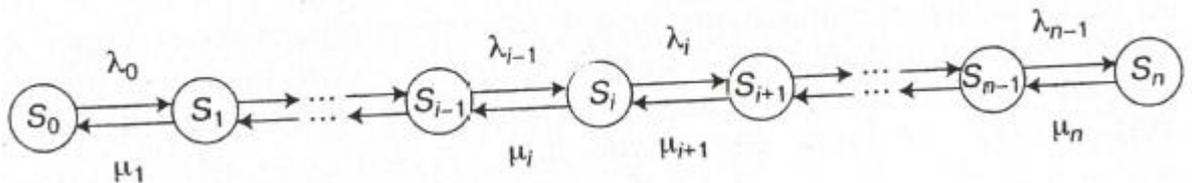


Рис. 3.8. Граф состояний

Предельные (финальные) вероятности состояний для простейшего эргодического процесса гибели и размножения, находящегося в стационарном режиме, определяются по следующим формулам:

$$P_k = \frac{\alpha_0 * \alpha_1 * \dots * \alpha_{k-1}}{\mu_1 * \mu_2 * \dots * \mu_k} * P_0, \quad k = 1, 2, \dots, n$$

$$P_0 = \left\{ 1 + \frac{\alpha_0}{\mu_1} + \frac{\alpha_0 * \alpha_1}{\mu_1 * \mu_2} + \dots + \frac{\alpha_0 * \alpha_1 * \dots * \alpha_{n-1}}{\mu_1 * \mu_2 * \dots * \mu_n} \right\}^{-1}$$

Правило. Вероятность k -го состояния в схеме гибели и размножения равна дроби, в числителе которой стоит произведение всех интенсивностей размножения, стоящих левее S_k , а в знаменателе – произведение всех интен-

сивностей гибели, стоящих левее S_k , умноженной на вероятность крайнего левого состояния системы P_0 .

В *примере 3.3* для стационарного режима если интенсивность поступления автомобилей постоянная ($\alpha(t) = \alpha = \text{const}$), то финальные вероятности состояний при условии, что нет ограничений на число автомобилей на предприятии, равны

$$P_0 = e^{-\frac{\alpha}{\mu}}$$

$$P_k = \left[\frac{\left(\frac{\alpha}{\mu}\right)^k}{k!} \right] * e^{-\frac{\alpha}{\mu}}$$

При этом математическое ожидание числа эксплуатируемых автомобилей равно его дисперсии:

$$M[A(t)] = D[A(t)] = \frac{\alpha}{\mu}$$

Если существует ограничение по числу автомобилей на предприятии (не более n), то финальные вероятности равны

$$P_0 = \frac{e^{-a}}{\sum_{k=0}^n a^k * e^{-\frac{a}{\mu}}}, \quad \text{где} \quad a = \frac{\alpha}{\mu}$$

$$P_k = P_0 * \frac{a^k}{k!}, \quad \text{где } k = 0, 1, 2, \dots, n$$

Математическое ожидание числа эксплуатируемых автомобилей в стационарном режиме

$$M[A(t)] = \sum_{k=0}^n k * P_k$$

3.3. Моделирование систем массового обслуживания

3.3.1. Компоненты и классификация моделей массового обслуживания

Системы массового обслуживания – это такие системы, в которые в случайные моменты времени поступают заявки на обслуживание, при этом

поступившие заявки обслуживаются с помощью имеющихся в распоряжении системы каналов обслуживания.

С позиции моделирования процесса массового обслуживания ситуации, когда образуются очереди заявок (требований) на обслуживание, возникают следующим образом. Поступив в обслуживающую систему, требование присоединяется к очереди других (ранее поступивших) требований. Канал обслуживания выбирает требование из находящихся в очереди, с тем чтобы приступить к его обслуживанию. После завершения процедуры обслуживания очередного требования канал обслуживания приступает к обслуживанию следующего требования, если таковое имеется в блоке ожидания.

Цикл функционирования системы массового обслуживания подобного рода повторяется многократно в течение всего периода работы обслуживающей системы. При этом предполагается, что переход системы на обслуживание очередного требования после завершения обслуживания предыдущего требования происходит мгновенно, в случайные моменты времени.

Примерами систем массового обслуживания могут служить: посты технического обслуживания автомобилей; посты ремонта автомобилей; персональные компьютеры, обслуживающие поступающие заявки или требования на решение тех или иных задач; станции технического обслуживания автомобилей; аудиторские фирмы; отделы налоговых инспекций, занимающиеся приемкой и проверкой текущей отчетности предприятий; телефонные станции и т. д.

Основными компонентами системы массового обслуживания любого вида являются: входной поток поступающих требований или заявок на обслуживание; дисциплина очереди; механизм обслуживания.

Входной поток требований. Для описания входного потока требуется задать вероятностный закон, определяющий последовательность моментов поступления требований на обслуживание и указать количество таких требований в каждом очередном поступлении.

Дисциплина очереди определяет принцип, в соответствии с которым поступающие на вход обслуживающей системы требования подключаются из очереди к процедуре обслуживания. Чаще всего используются дисциплины очереди, определяемые следующими правилами: первым пришел – первым обслуживаешься; пришел последним – обслуживаешься первым; случайный отбор заявок; отбор заявок по критерию приоритетности; ограниче-

ние времени ожидания момента наступления обслуживания (имеет место очередь с ограниченным временем ожидания обслуживания, что ассоциируется с понятием «допустимая длина очереди»).

Механизм обслуживания определяется характеристиками самой процедуры обслуживания и структурой обслуживающей системы. К характеристикам процедуры обслуживания относятся: продолжительность процедуры обслуживания; количество требований, удовлетворяемых в результате выполнения каждой такой процедуры; вероятность выхода обслуживающего прибора по истечении некоторого ограниченного интервала времени.

Структура обслуживающей системы определяется количеством и взаимным расположением каналов обслуживания (механизмов, приборов и т. п.). Прежде всего следует подчеркнуть, что система обслуживания может иметь не один канал обслуживания, а несколько. Система такого рода способна обслуживать одновременно несколько требований. В этом случае все каналы обслуживания предлагают одни и те же услуги, и, следовательно, можно утверждать, что имеет место параллельное обслуживание.

Система обслуживания может состоять из нескольких разнотипных каналов обслуживания, через которые должно пройти каждое обслуживаемое требование, т. е. в обслуживающей системе процедуры обслуживания требований реализуются последовательно. Механизм обслуживания определяет характеристики выходящего (обслуженного) потока требований.

Рассмотрев основные компоненты систем обслуживания, можно констатировать, что функциональные возможности любой системы массового обслуживания определяются следующими основными факторами:

- вероятностным распределением моментов поступлений заявок на обслуживание (единичных или групповых);
- вероятностным распределением времени продолжительности обслуживания;
- конфигурацией обслуживающей системы (параллельное, последовательное или параллельно-последовательное обслуживание);
- количеством и производительностью обслуживающих каналов;
- дисциплиной очереди;
- мощностью источника требований.

В качестве основных *критериев эффективности функционирования систем массового обслуживания* в зависимости от характера решаемой задачи могут выступать:

- вероятность немедленного обслуживания поступившей заявки;
- вероятность отказа в обслуживании поступившей заявки;
- относительная и абсолютная пропускная способность системы;
- средний процент заявок, получивших отказ в обслуживании;
- среднее время ожидания в очереди;
- средняя длина очереди;
- средний доход от функционирования системы в единицу времени и т.п.

Предметом теории массового обслуживания является установление зависимости между факторами, определяющими функциональные возможности системы массового обслуживания, и эффективностью ее функционирования. В большинстве случаев все параметры, описывающие системы массового обслуживания, являются случайными величинами или функциями, поэтому эти системы относятся к стохастическим системам.

Случайный характер потока заявок (требований), а также, в общем случае, и длительности обслуживания приводит к тому, что в системе массового обслуживания происходит случайный процесс. По характеру случайного процесса, происходящего в системе массового обслуживания (СМО), различают системы марковские и немарковские. В марковских системах входящий поток требований и выходящий поток обслуженных требований (заявок) являются пуассоновскими. Пуассоновские потоки позволяют легко описать и построить математическую модель системы массового обслуживания. Данные модели имеют достаточно простые решения, поэтому большинство известных приложений теории массового обслуживания используют марковскую схему. В случае немарковских процессов задачи исследования систем массового обслуживания значительно усложняются и требуют применения статистического моделирования, численных методов с использованием ЭВМ.

Независимо от характера процесса, протекающего в системе массового обслуживания, различают *два основных вида СМО*:

- системы с отказами, в которых заявка, поступившая в систему в момент, когда все каналы заняты, получает отказ и сразу же покидает очередь;
- системы с ожиданием (очередью), в которых заявка, поступившая в момент, когда все каналы обслуживания заняты, становится в очередь и ждет, пока не освободится один из каналов.

Системы массового обслуживания с ожиданием делятся на системы с ограниченным ожиданием и системы с неограниченным ожиданием.

В системах с ограниченным ожиданием может ограничиваться: длина очереди; время пребывания в очереди.

В системах с неограниченным ожиданием заявка, стоящая в очереди, ждет обслуживание неограниченно долго, т.е. пока не подойдет очередь.

Все системы массового обслуживания различают *по числу каналов обслуживания*: одноканальные системы; многоканальные системы.

Приведенная классификация СМО является условной. На практике чаще всего системы массового обслуживания выступают в качестве смешанных систем. Например, заявки ожидают начала обслуживания до определенного момента, после чего система начинает работать как система с отказами.

3.3.2. Определение характеристик систем массового обслуживания

Простейшей одноканальной моделью с вероятностными входным потоком и процедурой обслуживания является модель, характеризующаяся показательным распределением как длительностей интервалов между поступлениями требований, так и длительностей обслуживания. При этом плотность распределения длительностей интервалов между поступлениями требований имеет вид:

$$f_1(t) = \lambda \cdot e^{-\lambda t}$$

где λ - интенсивность поступления заявок в систему.

Плотность распределения длительностей обслуживания:

$$f_2(t) = \mu \cdot e^{-\mu t}$$

где μ - интенсивность обслуживания.

Потоки заявок и обслуживаний простейшие. Система работает с *отказами*.

Данная система массового обслуживания может быть представлена в виде графа (рис. 1.13), у которого имеются два состояния:

S_0 - канал свободен (ожидание);

S_1 - канал занят (идет обслуживание заявки).

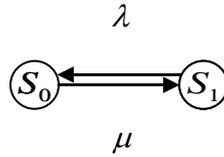


Рис. 3.9. Граф состояний одноканальной СМО с отказами

Обозначим вероятности состояний:

$P_0(t)$ — вероятность состояния «канал свободен»;

$P_1(t)$ — вероятность состояния «канал занят».

$$P_0(t) + P_1(t) = 1$$

$$P_1(t) = 1 - P_0(t)$$

$$P_0(t) = \frac{\lambda}{\lambda + \mu} e^{-(\lambda + \mu)t} + \frac{\mu}{\lambda + \mu}$$

Для одноканальной СМО с отказами вероятность $P_0(t)$ есть не что иное, как *относительная пропускная способность* системы q . Действительно, P_0 — вероятность того, что в момент t канал свободен и заявка, пришедшая к моменту t , будет обслужена, а следовательно, для данного момента времени t среднее отношение числа обслуженных заявок к числу поступивших также равно $P_0(t)$, то есть

$$q = P_0(t)$$

По истечении большого интервала времени (при $t \rightarrow \infty$) достигается стационарный (установившийся) режим:

$$q = P_0 = \frac{\mu}{\mu + \lambda}$$

Зная относительную пропускную способность, можно найти абсолютную. *Абсолютная пропускная способность* (A) — среднее число заявок, которое может обслужить система массового обслуживания в единицу времени:

$$A = \lambda \cdot q = \frac{\lambda \cdot \mu}{\lambda + \mu}$$

Вероятность отказа в обслуживании заявки будет равна вероятности состояния «канал занят»:

$$P_{\text{отк}} = P_1 = 1 - P_0 = 1 - \frac{\mu}{\lambda + \mu} = \frac{\lambda}{\lambda + \mu}$$

Данная величина $P_{\text{отк}}$ может быть интерпретирована как средняя доля необслуженных заявок среди поданных.

Пример 3.4. Пусть одноканальная СМО с отказами представляет собой один пост ежедневного обслуживания (ЕО) для мойки автомобилей. Заявка – автомобиль, прибывший в момент, когда пост занят – получает отказ в обслуживании. Интенсивность потока автомобилей $\lambda = 1,0$ (автомобиль в час). Средняя продолжительность обслуживания - 1,8 часа. Поток автомобилей и поток обслуживаний являются простейшими.

Требуется определить в установившемся режиме предельные значения: относительной пропускной способности q ; абсолютной пропускной способности A ; вероятности отказа $P_{\text{отк}}$.

Необходимо сравнить фактическую пропускную способность СМО с номинальной, которая была бы, если бы каждый автомобиль обслуживался точно 1,8 часа и автомобили следовали один за другим без перерыва.

Решение

1. Определим интенсивность потока обслуживания:

$$\mu = \frac{1}{t_{\text{об}}} = \frac{1}{1,8} = 0,555$$

2. Вычислим относительную пропускную способность:

$$q = \frac{\mu}{\lambda + \mu} = \frac{0,555}{1 + 0,555} = 0,356$$

Величина q означает, что в установившемся режиме система будет обслуживать примерно 35% прибывающих на пост ЕО автомобилей.

3. Абсолютную пропускную способность определим по формуле:

$$A = \lambda \cdot q = 1 \cdot 0,356 = 0,356$$

Это означает, что система (пост ЕО) способна осуществить в среднем 0,356 обслуживания автомобилей в час.

3. Вероятность отказа:

$$P_{\text{отк}} = 1 - q = 1 - 0,356 = 0,644$$

Это означает, что около 65% прибывших автомобилей на пост ЕО получат отказ в обслуживании.

4. Определим номинальную пропускную способность системы (автомобилей в час):

$$A_{\text{ном}} = \frac{1}{t_{\text{обсл}}} = \frac{1}{1,8} = 0,555$$

Оказывается, что в $A_{\text{ном}}$ в 1,5 раза $\left(\frac{0,555}{0,356} \approx 1,5\right)$ больше, чем фактическая пропускная способность, вычисленная с учетом случайного характера потока заявок и времени обслуживания.

Рассмотрим *одноканальную СМО с ожиданием*.

Система массового обслуживания имеет один канал. Входящий поток заявок - простейший поток с интенсивностью λ . Интенсивность потока обслуживания равна μ (т. е. в среднем непрерывно занятый канал будет выдавать μ обслуженных заявок). Длительность обслуживания – случайная величина, подчиненная показательному закону распределения. Поток обслуживаний является простейшим пуассоновским потоком событий. Заявка, поступившая в момент, когда канал занят, становится в очередь и ожидает обслуживания.

Предположим, что независимо от того, сколько требований подступает на вход обслуживающей системы, данная система (очередь + обслуживаемые клиенты) не может вместить более N -требований (заявок), т. е. клиенты, не попавшие в ожидание, вынуждены обслуживаться в другом месте. Источник, порождающий заявки на обслуживание, имеет неограниченную (бесконечно большую) емкость.

Граф состояний СМО в этом случае имеет вид, показанный на рис 1.14.

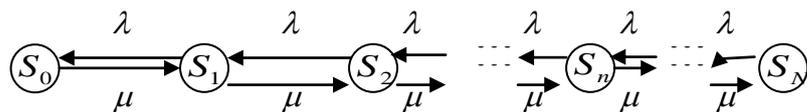


Рис. 3.10. Граф состояний одноканальной СМО с ожиданием (схема гибели и размножения)

Состояния СМО имеют следующую интерпретацию:

S_0 — «канал свободен»;

S_1 — «канал занят» (очереди нет);

S_2 — «канал занят» (одна заявка стоит в очереди);

.....

S_n — «канал занят» ($n - 1$ заявок стоит в очереди);

.....

S_N — «канал занят» ($N - 1$ заявок стоит в очереди).

Стационарный процесс в системе будет описываться системой алгебраических уравнений, решение которой для модели СМО имеет вид:

$$\text{где } \rho = \frac{\lambda}{\mu} \quad P_n = \begin{cases} P_0 \cdot \rho^n, \rho \neq 1, n = 1, 2, \dots, N \\ \frac{1}{(N+1)}, \rho = 1 \end{cases}$$

Определим характеристики одноканальной СМО с ожиданием и ограниченной длиной очереди, равной $(N-1)$:

- вероятность отказа в обслуживании заявки:

$$P_{\text{отк}} = P_N = \begin{cases} \left(\frac{1-\rho}{1-\rho^{N+1}} \right) \rho^N, \rho \neq 1 \\ \frac{1}{(N+1)}, \rho = 1 \end{cases}$$

- относительная пропускная способность системы:

$$q = 1 - P_{\text{отк}} = \begin{cases} 1 - \left(\frac{1-\rho}{1-\rho^{N+1}} \right) \rho^N, \rho \neq 1 \\ 1 - \frac{1}{(N+1)}, \rho = 1 \end{cases}$$

- абсолютная пропускная способность:

$$A = q \cdot \lambda$$

- среднее число находящихся в системе заявок:

$$L_s = \sum_{n=0}^N n \cdot P_n = \begin{cases} \frac{\rho \cdot [1 - (N+1) \cdot \rho^N + N \cdot \rho^{N+1}]}{(1-\rho) \cdot (1-\rho^{N+1})}, \rho \neq 1 \\ N/2, \rho = 1 \end{cases}$$

- среднее время пребывания заявки в системе:

$$W_s = \frac{L_s}{\lambda(1-P_N)}$$

- средняя продолжительность пребывания клиента (заявки) в очереди:

$$W_d = W_s - 1/\mu$$

- среднее число заявок (клиентов) в очереди (длина очереди)

$$L_q = \lambda(1 - P_N)W_d$$

Рассмотрим пример одноканальной СМО с ожиданием.

Пример 3.5. Специализированный пост диагностики представляет собой одноканальную СМО. Число стоянок для автомобилей, ожидающих проведения диагностики, ограничено и равно 3 [(N-1) = 3]. Если все стоянки заняты, т. е. в очереди уже находится три автомобиля, то очередной автомобиль, прибывший на диагностику, в очередь на обслуживание не становится. Поток автомобилей, прибывающих на диагностику, распределен по закону Пуассона и имеет интенсивность $\lambda = 0,85$ (автомобилей в час). Время диагностики автомобиля распределено по показательному закону и в среднем равно 1,05 час.

Требуется определить вероятностные характеристики поста диагностики, работающего в стационарном режиме.

Решение

1. Параметр потока обслуживаний автомобилей:

$$\mu = \frac{1}{t} = \frac{1}{1,05} = 0,952$$

2. Приведенная интенсивность потока автомобилей определяется как отношение интенсивностей λ и μ , т. е.

$$\rho = \frac{\lambda}{\mu} = \frac{0,85}{0,952} = 0,893$$

3. Вычислим финальные вероятности системы:

$$P_0 = \frac{1 - \rho}{1 - \rho^{N+1}} = \frac{1 - 0,893}{1 - 0,893^5} \approx 0,248$$

$$P_1 = \rho \cdot P_0 = 0,893 \cdot 0,248 \approx 0,221$$

$$P_2 = \rho^2 \cdot P_0 = 0,893^2 \cdot 0,248 \approx 0,198$$

$$P_3 = \rho^3 \cdot P_0 = 0,893^3 \cdot 0,248 \approx 0,177$$

$$P_4 = \rho^4 \cdot P_0 = 0,893^4 \cdot 0,248 \approx 0,158$$

4. Вероятность отказа в обслуживании автомобиля:

$$P_{отк} = P_4 = \rho^4 \cdot P_0 \approx 0,158$$

5. Относительная пропускная способность поста диагностики:

$$q = 1 - P_{отк} = 1 - 0,158 = 0,842$$

1. Абсолютная пропускная способность поста диагностики (автомобиля в час)

$$A = \lambda \cdot q = 0,85 \cdot 0,842 = 0,716$$

7. Среднее число автомобилей, находящихся на обслуживании и в очереди (т.е. в системе массового обслуживания):

$$L_s = \frac{\rho \cdot [1 - (N+1) \cdot \rho^N + N \cdot \rho^{N+1}]}{(1-\rho) \cdot (1-\rho^{N+1})} = \frac{0,893 \cdot [1 - (4+1) \cdot 0,893^4 + 4 \cdot 0,893^5]}{(1-0,893) \cdot (1-0,893^5)} = 1,77$$

8. Среднее время пребывания автомобиля в системе (час):

$$W_s = \frac{L_s}{\lambda(1-P_N)} = \frac{1,77}{0,85(1-0,158)} \approx 2,473$$

9. Средняя продолжительность пребывания заявки в очереди на обслуживание (час):

$$W_q = W_s - 1/\mu = 2,473 - 1/0,952 = 1,423$$

10. Среднее число заявок в очереди (длина очереди):

$$L_q = \lambda \cdot (1 - P_N) \cdot W_q = 0,85 \cdot (1 - 0,158) \cdot 1,423 = 1,02$$

Работу рассмотренного поста диагностики можно считать удовлетворительной, так как пост диагностики не обслуживает автомобили в среднем в 15,8% случаев ($P_{отк} = 0,158$).

Рассмотрим *одноканальную СМО с ожиданием без ограничения на вместимость блока ожидания* (т. е. $N \rightarrow \infty$). Остальные условия функционирования СМО остаются без изменений.

Характеристики одноканальной СМО с ожиданием, без ограничения на длину очереди, следующие:

- среднее число находящихся в системе клиентов (заявок) на обслуживание:

$$L_s = \sum_{n=0}^{\infty} n \cdot P_n = \frac{\rho}{1-\rho}$$

- средняя продолжительность пребывания клиента в системе:

$$W_s = \frac{L_s}{\lambda} = \frac{1}{[\mu \cdot (1 - \rho)]}$$

- среднее число клиентов в очереди на обслуживании:

$$L_q = L_s - \frac{\lambda}{\mu} = \frac{\rho^2}{(1 - \rho)}$$

- средняя продолжительность пребывания клиента в очереди:

$$W_q = \frac{L_q}{\lambda} = \frac{\rho}{[\mu \cdot (1 - \rho)]}$$

Пример 3.6. Специализированный пост диагностики представляет собой одноканальную СМО. Пост диагностики располагает неограниченным количеством площадок для стоянки прибывающих на обслуживание автомобилей, т. е. длина очереди не ограничена. Поток автомобилей, прибывающих на диагностику, распределен по закону Пуассона и имеет интенсивность $\lambda = 0,85$ (автомобиля в час). Время диагностики автомобиля распределено по показательному закону и в среднем равно 1,05 час.

Требуется определить финальные значения следующих вероятностных характеристик: вероятности состояний системы (поста диагностики); среднее число автомобилей, находящихся в системе (на обслуживании и в очереди); среднюю продолжительность пребывания автомобиля в системе (на обслуживании и в очереди); среднее число автомобилей в очереди на обслуживании; среднюю продолжительность пребывания автомобиля в очереди.

Решение

1. Параметр потока обслуживания μ и приведенная интенсивность потока автомобилей ρ определены в примере 1.10:

$$\text{а. } \mu = 0,952; \rho = 0,893.$$

2. Вычислим предельные вероятности системы по формулам

$$P_0 = 1 - \rho = 1 - 0,893 = 0,107;$$

$$P_1 = (1 - \rho) \cdot \rho = (1 - 0,893) \cdot 0,893 = 0,096;$$

$$P_2 = (1 - \rho) \cdot \rho^2 = (1 - 0,893) \cdot 0,893^2 = 0,085;$$

$$P_3 = (1-\rho) \cdot \rho^3 = (1 - 0,893) \cdot 0,893^3 = 0,076;$$

$$P_4 = (1-\rho) \cdot \rho^4 = (1 - 0,893) \cdot 0,893^4 = 0,068;$$

$$P_5 = (1-\rho) \cdot \rho^5 = (1 - 0,893) \cdot 0,893^5 = 0,061 \text{ и т. д.}$$

Следует отметить, что P_0 определяет долю времени, в течение которого пост диагностики вынужденно бездействует (простаивает). В нашем примере она составляет 10,7%, так как $P_0 = 0,107$.

3. Среднее число автомобилей, находящихся в системе (на обслуживании и в очереди):

$$L_s = \frac{\rho}{1-\rho} = \frac{0,893}{1-0,893} = 8,346$$

4. Средняя продолжительность пребывания клиента в системе (час):

$$W_s = \frac{L_s}{\lambda} = \frac{1}{[\mu \cdot (1-\rho)]} = \frac{1}{[0,952 \cdot (1-0,893)]} = 9,817$$

5. Среднее число автомобилей в очереди на обслуживание:

$$L_q = L_s - \frac{\lambda}{\mu} = \frac{\rho^2}{(1-\rho)} = \frac{0,893^2}{(1-0,893)} = 7,453$$

6. Средняя продолжительность пребывания автомобиля в очереди (час):

$$W_q = \frac{\rho}{\mu \cdot (1-\rho)} = \frac{0,893}{0,952 \cdot (1-0,893)} = 8,766$$

7. Относительная пропускная способность системы:

$$q = 1$$

т. е. каждая заявка, пришедшая в систему, будет обслужена.

8. Абсолютная пропускная способность:

$$A = \lambda \cdot q = 0,85 \cdot 1 = 0,85$$

В подавляющем большинстве случаев на практике системы массового обслуживания являются многоканальными, и, следовательно, **модели с n обслуживающими каналами** (где $n > 1$) представляют несомненный интерес.

Процесс массового обслуживания, описываемый данной моделью, характеризуется интенсивностью входного потока λ , при этом параллельно может обслуживаться не более n клиентов (заявок). Средняя продолжитель-

ность обслуживания одной заявки равняется $1/\mu$. Входной и выходной потоки являются пуассоновскими. Режим функционирования того или иного обслуживающего канала не влияет на режим функционирования других обслуживающих каналов системы, причем длительность процедуры обслуживания каждым из каналов является случайной величиной, подчиненной экспоненциальному закону распределения. Конечная цель использования n параллельно включенных обслуживающих каналов заключается в повышении (по сравнению с одноканальной системой) скорости обслуживания требований за счет обслуживания одновременно n клиентов.

Граф состояний многоканальной системы массового обслуживания с отказами имеет вид, показанный на рис. 3.11.

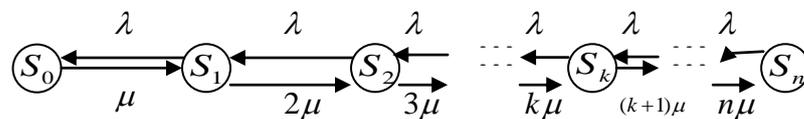


Рис. 3.11. Граф состояний многоканальной СМО с отказами

Состояния данной СМО имеют следующую интерпретацию:

S_0 — все каналы свободны;

S_1 — занят один канал, остальные свободны;

.....

S_k — заняты ровно k каналов, остальные свободны;

.....

S_n — заняты все n каналов, заявка получает отказ в обслуживании.

Начальные условия решения системы таковы:

$$P_0(0) = 1, P_1(0) = P_2(0) = \dots = P_k(0) = \dots = P_n(0) = 0$$

Стационарное решение системы имеет вид:

$$\left\{ \begin{array}{l} P_k = \frac{\rho^k}{k!} = \frac{\rho}{k!} \cdot P_0, k = 0, 1, 2, \dots, n \\ P_0 = \frac{1}{\left[\sum_{k=0}^n \frac{\rho^k}{k!} \right]}, k = 0, 1, 2, \dots, n \end{array} \right.$$

где $\rho = \frac{\lambda}{\mu}$

Формулы для вычисления вероятностей P_k называются **формулами Эрланга**.

Вероятностные характеристики функционирования многоканальной СМО с отказами в стационарном режиме:

- вероятность отказа (заявка получает отказ, если приходит в момент, когда все n каналов заняты. Величина $P_{отк}$ характеризует полноту обслуживания входящего потока):

$$P_{отк} = P_n = \frac{\rho^n}{n!} \cdot P_0,$$

- вероятность того, что заявка будет принята к обслуживанию (она же — относительная пропускная способность системы q) дополняет $P_{отк}$ до единицы:

$$q = 1 - P_{отк} = 1 - \frac{\rho^n}{n!} \cdot P_0,$$

- абсолютная пропускная способность:

$$A = \lambda \cdot q = \lambda \cdot (1 - P_{отк})$$

- среднее число каналов, занятых обслуживанием (\bar{k}) следующее:

$$\bar{k} = \sum_{k=1}^n k \cdot P_k = \rho \cdot (1 - P_{отк})$$

Величина \bar{k} характеризует степень загрузки СМО.

Пример 3.7. Пусть n -канальная СМО представляет собой вычислительный центр (ВЦ) с тремя ($n = 3$) взаимозаменяемыми ПЭВМ для решения поступающих задач. Поток задач, поступающих на ВЦ, имеет интенсивность $\lambda = 1$ задаче в час. Средняя продолжительность обслуживания $\bar{t}_{обсл} = 1,8$ час. Поток заявок на решение задач и поток обслуживания этих заявок являются простейшими.

Требуется вычислить финальные значения: вероятности состояний ВЦ; вероятности отказа в обслуживании заявки; относительной пропускной способности ВЦ; абсолютной пропускной способности ВЦ; среднего числа занятых ПЭВМ на ВЦ.

Решение

1. Определим параметр μ потока обслуживаний:

$$\mu = \frac{1}{\bar{t}_{\text{обсл}}} = \frac{1}{1,8} = 0,555$$

2. Приведенная интенсивность потока заявок:

$$\rho = \lambda / \mu = 1 / 0,555 = 1,8$$

3. Предельные вероятности состояний найдем по формулам Эрланга:

$$P_1 = \frac{\rho}{1!} \cdot P_0 = 1,8 \cdot P_0$$

$$P_2 = \frac{\rho^2}{2!} \cdot P_0 = 1,62 \cdot P_0$$

$$P_3 = \frac{\rho^3}{3!} \cdot P_0 = 0,97 \cdot P_0$$

$$P_0 = \frac{1}{\sum_{k=0}^3 \frac{\rho^k}{k!}} = \frac{1}{1 + 1,8 + 1,62 + 0,97} \approx 0,186$$

$$P_1 \approx 1,8 \cdot 0,186 \approx 0,334$$

$$P_2 \approx 1,62 \cdot 0,186 \approx 0,301$$

$$P_3 \approx 0,97 \cdot 0,186 \approx 0,180$$

4. Вероятность отказа в обслуживании заявки:

$$P_{\text{отк}} = P_3 = 0,180$$

5. Относительная пропускная способность ВЦ:

$$q = 1 - P_{\text{отк}} = 1 - 0,180 = 0,820$$

6. Абсолютная пропускная способность ВЦ:

$$A = \lambda \cdot q = 1 \cdot 0,820 = 0,820$$

7. Среднее число занятых каналов — ПЭВМ:

$$\bar{k} = \rho \cdot (1 - P_{\text{отк}}) = 1,8 \cdot (1 - 0,180) = 1,476$$

Таким образом, при установившемся режиме работы СМО в среднем будет занято 1,5 компьютера из трех — остальные полтора будут простаивать. Работу рассмотренного ВЦ вряд ли можно считать удовлетворительной, так как центр не обслуживает заявки в среднем в 18% случаев (P_3 — 0,180). Очевидно, что пропускную способность ВЦ при данных λ и μ можно увеличить только за счет увеличения числа ПЭВМ.

Рассмотрим *многоканальную систему массового обслуживания с ожиданием*. Процесс массового обслуживания при этом характеризуется следующим: входной и выходной потоки являются пуассоновскими с интенсивностями λ и μ соответственно; параллельно обслуживаться могут не более C клиентов. Система имеет C каналов обслуживания. Средняя продолжительность обслуживания одного клиента равна $1/\mu$.

В установившемся режиме функционирование многоканальной СМО с ожиданием и неограниченной очередью может быть описано с помощью системы алгебраических уравнений, решение которой имеет вид:

$$\begin{cases} P_n = \frac{\rho^n}{n!} \cdot P_0 & 0 \leq n \leq C \\ P_n = \frac{\rho^n}{C! \cdot C^{n-C}} \cdot P_0 & n \geq C \end{cases}$$

где

$$P_0 = \left\{ \sum_{n=0}^{C-1} \frac{\rho^n}{n!} + \frac{\rho^C}{C! \left[1 - \left(\frac{\rho}{C} \right) \right]} \right\}^{-1}$$

Вероятностные характеристики функционирования в стационарном режиме многоканальной СМО с ожиданием и неограниченной очередью определяются по следующим формулам:

- вероятность того, что в системе находится n клиентов на обслуживании, определяется по формулам:

$$\begin{cases} P_n = \frac{\rho^n}{n!} \cdot P_0 & 0 \leq n \leq C \\ P_n = \frac{\rho^n}{C! \cdot C^{n-C}} \cdot P_0 & n \geq C \end{cases}$$

- среднее число клиентов в очереди на обслуживание

$$L_q = \left[\frac{C \cdot \rho}{(C \cdot \rho)^2} \right] \cdot P_C$$

- среднее число находящихся в системе клиентов (заявок на обслуживание и в очереди)

$$L_s = L_q + \rho$$

- средняя продолжительность пребывания клиента (заявки на обслуживание) в очереди

$$W_q = \frac{L_q}{\lambda}$$

- средняя продолжительность пребывания клиента в системе

$$W_s = W_q + \frac{1}{\mu}$$

Рассмотрим примеры многоканальной системы массового обслуживания с ожиданием.

Пример 3.8. Механическая мастерская завода с тремя постами (каналами) выполняет ремонт малой механизации. Поток неисправных механизмов, прибывающих в мастерскую, — пуассоновский и имеет интенсивность $\lambda = 2,5$ механизма в сутки, среднее время ремонта одного механизма распределено по показательному закону и равно $\bar{t} = 0,5$ сут. Предположим, что другой мастерской на заводе нет, и, значит, очередь механизмов перед мастерской может расти практически неограниченно.

Требуется вычислить следующие предельные значения вероятностных характеристик системы: вероятности состояний системы; среднее число заявок в очереди на обслуживание; среднее число находящихся в системе заявок; среднюю продолжительность пребывания заявки в очереди; среднюю продолжительность пребывания заявки в системе.

Решение

1. Определим параметр потока обслуживаний:

$$\mu = \frac{1}{t} = 1/0,5 = 2$$

2. Приведенная интенсивность потока заявок:

$$\rho = \lambda / \mu = 2,5 / 2,0 = 1,25$$

при этом $\lambda / \mu \cdot c = 2,5 / 2 \cdot 3 = 0,41$

Поскольку $\lambda / \mu \cdot c < 1$, то очередь не растет безгранично и в системе наступает предельный стационарный режим работы.

3. Вычислим вероятности состояний системы:

$$P_0 = \left\{ \sum_{n=0}^{C-1} \frac{\rho^n}{n!} + \frac{\rho^C}{C! \left[1 - \left(\frac{\rho}{C} \right) \right]} \right\}^{-1} = \frac{1}{\left[1 + \frac{\rho^1}{1!} + \frac{\rho^2}{2!} + \frac{\rho^3}{3! \left(1 - \frac{\rho}{3} \right)} \right]} = \frac{1}{\left[1 + \rho + \frac{\rho^2}{2} + \frac{\rho^3}{6 \left(1 - \frac{\rho}{3} \right)} \right]} =$$

$$= \frac{1}{\left[1 + 1,25 + \frac{1,25^2}{2} + \frac{1,25^3}{6 \left(1 - \frac{1,25}{3} \right)} \right]} = 0,279$$

$$P_1 = \frac{\rho^1}{1!} \cdot P_0 = 1,25 \cdot 0,279 = 0,349$$

$$P_2 = \frac{\rho^2}{2!} \cdot P_0 = \frac{1,25^2}{2!} \cdot 0,279 = 0,218$$

$$P_3 = \frac{\rho^3}{3!} \cdot P_0 = \frac{1,25^3}{3!} \cdot 0,279 = 0,091$$

$$P_4 = \frac{\rho^4}{4!} \cdot P_0 = \frac{1,25^4}{4!} \cdot 0,279 = 0,028$$

4. Вероятность отсутствия очереди у мастерской:

$$P_{om.o} = P_0 + P_1 + P_2 + P_3 \approx 0,279 + 0,349 + 0,218 + 0,091 = 0,937$$

5. Среднее число заявок и очереди ни обслуживание:

$$L_q = \left[\frac{C \cdot \rho}{(C \cdot \rho)^2} \right] \cdot P_C = \left[\frac{3 \cdot 1,25}{(3 \cdot 1,25)^2} \right] \cdot 0,091 = 0,111$$

6. Среднее число находящихся в системе заявок:

$$L_s = L_q + \rho = 0,111 + 1,25 = 1,361$$

7. Средняя продолжительность пребывания механизма в очереди на

$$\text{обслуживание (суток): } W_q = \frac{L_q}{\lambda} = \frac{0,111}{2,5} = 0,044$$

8. Средняя продолжительность пребывания механизма в мастерской (суток):

$$W_s = W_q + \frac{1}{\mu} = 0,044 + \frac{1}{2} = 0,544$$

Литература

- 1) Бережная Е.В. Математические методы моделирования экономических систем: учеб. пособие/ Е. В. Бережная, В. И. Березной. - изд. 2-е, перераб. и доп. - М: Финансы и статистика, 2008. – 430 с.
- 2) Верников Г. 7 тонн менеджмента [Электронный ресурс]. – Режим доступа: <http://vernikov.ru/krisis/item/33--idef1x.html>, свободный
- 3) Верников Г. Основные методологии обследования организаций. Стандарт IDEF0. [Электронный ресурс]. – Режим доступа: <http://www.interface.ru/fset.asp?Url=/ca/idefo.htm> , свободный
- 4) Высшая математика для экономистов / под ред. проф. Н. Ш. Кремера .— 3-е изд .— М.: ЮНИТИ-ДАНА, 2010 .— 478 с.
- 5) Глухов В.В., Медников М.Д., Коробко С.Б. Математические методы и модели для менеджмента. 3-е изд. – СПб.: Издательство «Лань», 2007. – 528 с.
- 6) Калянов Г. Н. Моделирование, анализ, реорганизация и автоматизация бизнес-процессов: учеб. Пособие. — М.: Финансы и статистика, 2007. — 240 с.
- 7) Калянов Г.Н. Консалтинг при автоматизации предприятий: подходы, методы, средства. – М.: СИНТЕГ, 2000
- 8) Ковалев С. М., Ковалев В. М. Бизнес-процессы, основные стандарты их описания // Справочник экономиста. — 2006. — №11
- 9) Кондратьев В. В., Кузнецов М. Н. и др. Показываем бизнес-процессы. — М.: Эксмо, 2007. — 352 с.
- 10) Маклаков С.В. Моделирование бизнес-процессов с BPwin 4.0. – Изд. «Диалог-МИФИ», 2009. - 224 с.
- 11) Методология функционального моделирования IDEF0. Руководящий документ [Электронный ресурс]. – Режим доступа: <http://www.nsu.ru/smk/files/idef.pdf> , свободный
- 12) Новиков М.В. IDEF0 в моделировании бизнес-процессов управления. [Электронный ресурс]. – Режим доступа: <http://idefinfo.ru/content/view/21/27/> , свободный
- 13) Репин В.В., Елиферов В.Г. Процессный подход к управлению. Моделирование бизнес-процессов. – Изд. «Стандарты и качество», 2009. – 408с.
- 14) Репин В.В.. Сравнительный анализ нотаций. [Электронный ресурс]. – Режим доступа: <http://www.interface.ru/fset.asp?Url=/ca/an/danaris1.htm> , свободный
- 15) Сериков А.В., Титов Н.В. Компьютерное моделирование бизнес-процессов. – Изд. «Бурун Книга», 2007. – 304 с.

- 16) Цуканова О.А. Математические методы моделирования экономических систем - СПб.: СПб ГУИТМО, 2012. – Электронное учебное пособие - Режим доступа: C:\fakepath\ПОСОБИЕ_MM_1.pdf
- 17) Черемных С. В. и др. Структурный анализ систем: IDEF-технологии. — М.: Финансы и статистика, 2001. — 208 с.
- 18) Электронно-библиотечная система. Издательство «Лань» [Электронный ресурс]. – Наливайко Л.В., Ивашина Н.В., Шмидт Ю.Д. Математика для экономистов. Сборник заданий – Лань, 2011. – Режим доступа: http://e.lanbook.com/books/element.php?pl1_cid=25&pl1_id=662
- 19) Information Integration for Concurrent Engineering (ИИСЕ) Compendium Of Methods Report [Электронный ресурс]. – Режим доступа: <https://www.cs.tcd.ie/Andrew.Butterfield/Teaching/CS4098/IDEF/IDEF-compendium.pdf> , свободный
- 20) Information Integration for Concurrent Engineering (ИИСЕ) IDEF4 Object-oriented Design Method Report [Электронный ресурс]. – Режим доступа: <http://www.idef.com/pdf/Idef4.pdf> , свободный
- 21) Information Integration for Concurrent Engineering (ИИСЕ) IDEF5 Method Report [Электронный ресурс]. – Режим доступа: <http://www.idef.com/pdf/Idef5.pdf> , свободный
- 22) Information Integration For Concurrent Engineering IDEF 3 Process Description Capture Method Report [Электронный ресурс]. – Режим доступа: <https://www.cs.tcd.ie/Andrew.Butterfield/Teaching/CS4098/IDEF/IDEF-compendium.pdf> , свободный

Миссия университета – генерация передовых знаний, внедрение инновационных разработок и подготовка элитных кадров, способных действовать в условиях быстро меняющегося мира и обеспечивать опережающее развитие науки, технологий и других областей для содействия решению актуальных задач.

КАФЕДРА ПРИКЛАДНОЙ ЭКОНОМИКИ И МАРКЕТИНГА

Кафедра прикладной экономики и маркетинга была создана в 1995 году в связи с реорганизацией кафедры экономики промышленности и организации производства. С момента основания кафедру возглавляет доктор экономических наук, профессор Олег Валентинович Васюхин.

С 1997 года кафедрой ПЭиМ проводилась подготовка экономистов по специальности 071900 «Информационные системы в экономике», а также бакалавров по направлению 521600 «Экономика». В настоящее время в связи с внедрением в учебный процесс стандартов нового поколения кафедра осуществляет подготовку специалистов по специальности 080801.65 «Прикладная информатика в экономике», бакалавров по направлению 080100.62 «Экономика», магистров по направлению 080100.68 «Экономика», магистерская программа «Экономика предпринимательской деятельности» и магистров по направлению 080500.68 «Бизнес-информатика», магистерская программа «IT-консалтинг».

С момента основания кафедры подготовлено в общей сложности более 600 специалистов, магистров и бакалавров. Выпускники кафедры имеют высокий рейтинг на рынке труда Санкт-Петербурга, что снимает проблемы с трудоустройством после окончания университета.

Преподаватели кафедры подготовили учебно-методическое обеспечение и ведут учебный процесс по таким дисциплинам, как «Информатика», «Информационные технологии», «Имитационное моделирование экономических процессов», «Экономика предприятия», «Внутрифирменное планирование», «Экономика информатики», «Экономика защиты информации», «Экономика и социология труда», «Сетевая экономика», «Маркетинг», «Предметно-ориентированные экономические информационные системы», «Экономика рынка недвижимости», «Управление проектами» и др.

Кафедра разрабатывает учебно-методические пособия. За последние несколько лет издано более 20 пособий, в частности, «Экономика предприятия и маркетинг», «Основы ценообразования», «Сетевая экономика», «Введение в программирование», «Офисное программирование» и др.

Обучение современным информационным технологиям проводится на основе материально-технической базы Гуманитарного факультета (ГФ). Компьютерные классы межкафедральной лаборатории ГФ и собственные ресурсы кафедры ПЭиМ насчитывают более 40 компьютеров и рабочих станций. Используется и лабораторная база других кафедр университета с имеющейся у них новейшей вычислительной, аудио- и видеотехникой.

Кафедра ведет международную научно-педагогическую деятельность, в частности, участвует в долгосрочной программе сотрудничества с Пекинским Механическим институтом в области перспектив экономического развития отраслей народного хозяйства, в рамках которой проводится обучение на кафедре ПЭиМ китайских студентов по направлению «Экономика».

Кафедра также осуществляет разветвлённую прикладную научную деятельность, возглавляемую и координируемую профессором Васюхиным О. В., специалистом в области организации производственных структур, на счету которого 63 опытно-конструкторских разработки, одна из которых удостоена бронзовой медали ВДНХ в 1982 году.

Один из важных аспектов кафедральной деятельности - интенсивная научная работа коллектива кафедры. Научную школу кафедры основал в 1975 году доктор экономических наук, профессор Владимир Арсентьевич Петров, выдающийся учёный советского периода, основоположник теории организации группового производства, являвшийся в то время членом Диссертационных советов многих Ленинградских вузов, председателем секции экономики и управления в ЛДНТП, участник международных конференций, книги которого были переведены и издавались в Италии, Болгарии, ГДР и других странах. В настоящее время научная школа профессора В.А. Петрова развивается за счет научных исследований и разработок преподавателей кафедры. За последние несколько лет было подготовлено и защищено 15 кандидатских и 2 докторских диссертации.

В результате обширной научной деятельности кафедра установила и поддерживает эффективное сотрудничество с аналогичными кафедрами СПбГУ, СПбГЭУ, СПб НИУ ВШЭ, СПбГУКиТ, СПбГУТиД, РЭУ им. Г.В. Плеханова и др.

Цуканова Ольга Анатольевна

**Методология и инструментарий моделирования бизнес-
процессов**

Учебное пособие

В авторской редакции
Редакционно-издательский отдел Университета ИТМО
Зав. РИО Н.Ф. Гусарова
Подписано к печати
Заказ №
Тираж 100 экз.
Отпечатано на ризографе