

## **Содержание:**

# **ВВЕДЕНИЕ**

Обработка данных современного предприятия любого размера и формы собственности немислима без использования компьютерных технологий и информационных систем (ИС).

Проектирование информационных систем в наши дни является весьма трудоемкой задачей, требующей времени и высокой квалификации участвующих в проектировании специалистов. Сложность разработки ИС обуславливается еще и тем, что потребности заказчика могут меняться в течение всего периода проектирования, тем, что при разработке системы возникает необходимость согласования проектных решений между различными группами специалистов и представителями заказчика. В связи с этим разработаны определенные спецификации и соглашения, стандартизирующие процессы проектирования информационных систем.

В основе проектирования информационных систем лежит моделирование предметной области. Для того чтобы получить адекватный предметной области проект ИС в виде системы правильно работающих программ, необходимо иметь целостное, системное представление о модели, которое отражает все аспекты функционирования будущей информационной системы. При этом под моделью предметной области понимается некоторая система, имитирующая структуру или функционирование исследуемой предметной области и отвечающая основному требованию – быть адекватной этой области.

Предварительное моделирование предметной области позволяет сократить время и сроки проведения проектировочных работ и получить более эффективный и качественный проект. Вследствие этого все современные технологии проектирования информационных систем основываются на использовании определенной методологии моделирования предметной области.

Актуальность данной курсовой работы обусловлена тем, что основным подходом при моделировании предметной области был структурный подход. Сущность структурного метода заключается в декомпозиции функций, проектируемой системы. Процесс разбиения (декомпозиции) на функции происходит до тех пор,

пока не получаем функцию, которая может быть легко запрограммирована с помощью выбранного алгоритмического языка. Такое проектирование "сверху вниз" позволяет не упускать из виду всю систему в целом.

Структурный метод имеет более чем 50-летнюю историю создания, однако в наше время использование функционального подхода и декомпозиции при моделировании предметной области позволяет представить проектируемую систему более наглядно, в связи с чем изучение методик структурного моделирования является актуальной задачей.

Объект курсовой работы: структурный подход к проектированию информационных систем.

Предмет данной курсовой работы: методология структурного проектирования.

Целью данной курсовой работы является изучение и анализ средств реализации структурных методов анализа и проектирования экономической информационной системы.

В рамках структурного подхода был разработан ряд методик, которые будут рассмотрены в курсовой работе.

Задачи курсовой работы:

- рассмотреть основные понятия проектирования информационных систем и подходы к проектированию информационных систем;
- изучить наиболее популярные структурные методологии проектирования информационных систем и их графическое представление;
- провести сравнительный анализ средств реализации структурных методов анализа и проектирования информационных систем.

Источниками для написания данной курсовой работы являются учебные пособия, а также издания авторов мнения, которых имеет вес в научной сфере.

## **Глава 1. Структурный подход к проектированию информационных систем**

# 1.1 Основные понятия проектирования информационных систем

Информационно-технологический фактор в наше время выступает в качестве ключевого в управлении предприятием любой формы собственности и отрасли экономики. Сущность этого фактора трактуется на современном этапе как создание единого информационного пространства управления предприятием и обеспечение возможности его использования для принятия решения в реальном времени. Реальным выражением этого фактора являются информационные системы предприятия, сложность которых с течением времени все возрастает. [13]

Информация, циркулирующая в любой социально-экономической системе, представляет информационную систему (информационную среду), характеризующуюся потоками, многочисленными обменами и преобразованиями, взаимосвязями. Совокупность объекта управления и системы управления образует информационную систему (ИС) (рис. 1).

Система управления

Объект управления

Входные информационные потоки

Выходные информационные потоки

Управляющие информационные потоки

Внешние информационные потоки

## Рисунок 1. Информационные потоки в ИС

Информационная система представляет собой коммуникационную систему по сбору, передаче, переработке информации об объекте, снабжающую работника любой профессии информацией для реализации функции управления. Другими словами информационная система - это упорядоченная совокупность документированной информации и информационных технологий.

Особенности развития современных информационных технологий характеризуются все возрастающей сложностью современных информационных систем, которые характеризуются следующими особенностями:

- большое количество функций, процессов, элементов данных, требующих тщательного описания и отладки;
- необходимость интеграции уже существующих и разрабатываемых систем;
- разработка системы для нескольких аппаратных (и, возможно, программных) платформ;
- разобщенность отдельных групп разработчиков и их разная профессиональная подготовка;
- отсутствие полных аналогов разрабатываемой системе и, тем самым, невозможность использования типовых решений;
- значительная временная протяженность разработки обусловленная большим объемом кода в разрабатываемой системе, ограниченностью людских ресурсов, масштабом организации-заказчика.[18]

Накопленный к настоящему времени опыт создания систем программного обеспечения (ПО) показывает, что это логически сложная, трудоемкая и длительная работа, требующая высокой квалификации участвующих в ней специалистов. Кроме того, в процессе создания и функционирования ПО потребности пользователей постоянно изменяются или уточняются, что еще более усложняет разработку и сопровождение таких систем.[7]

В основе проектирования информационных систем лежит моделирование предметной области. Чтобы получить адекватный предметной области проект ИС как систему корректно работающих программ, необходимо иметь системное и целостное представление модели, которое отражает все аспекты функционирования разрабатываемой ИС.[19]

Моделирование ИС — процесс разработки модели предметной области, а также процесс использования модели для получения количественно-качественных характеристик какого-либо процесса или явления.[22]

Функциями моделирования ИС является: описание, объяснение и прогнозирование поведения реальной системы, а также улучшение бизнес-процессов.

При создании информационной системы важным решением является выбор и обоснование методологии разработки программных систем. Подходы к проектированию ИС различаются между собой критериями декомпозиции:

1. Структурный подход. Система разбивается на подсистемы до выделения функциональных элементов, между которыми устанавливается строгий порядок выполняемых действий.

2. Объектно-ориентированный. Выделяются объекты, содержательные данные и методы их обработки. Объекты обладают характерным для них поведением и, взаимодействуя друг с другом, обеспечивают общее поведение системы.[3]

Структурный подход при моделировании предметной области был первым и долгое время основным.

## **1.2 Методологии структурного подхода к проектированию информационных систем**

Сущность структурного подхода к проектированию информационных систем заключается в декомпозиции (разбиении) системы на автоматизируемые функции (функциональные подсистемы), которые, в свою очередь, делятся на подфункции, подразделяемые на задачи, и т.д. Процесс разбиения функций продолжается вплоть до конкретных процедур. Автоматизируемая система при этом сохраняет целостное представление, где все взаимоувязаны друг с другом составляющие компоненты.[11]

В случае разработки системы «снизу вверх», то есть от конкретных задач к системе, ее целостность теряется, могут возникать проблемы при информационной стыковке отдельных компонент.

Все наиболее распространенные методологии структурного подхода базируются на ряде общих принципов:

- разбиение общей проблемы на множество меньших независимых подзадач;
- принцип организации составных частей проблемы в иерархические древовидные структуры;
- принцип строгого методического подхода к решению проблемы.[6]

Современные структурные методологии анализа и проектирования классифицируются по следующим признакам (таблица 1):

**Таблица 1.** Классификация структурных методологий анализа и проектирования ИС

Признак классификации	Виды методологий	Краткая характеристика
по отношению к школам	<ul style="list-style-type: none"> <li>- Software Engineering (SE);</li> <li>- Information Engineering (IE)</li> </ul>	<p>IE используется только для построения информационных систем, а SE - для всех типов систем</p>
по порядку построения модели	<ul style="list-style-type: none"> <li>- процедурно-ориентированные;</li> <li>- ориентированные на данные;</li> <li>- информационно-ориентированные</li> </ul>	<p>Процедурно-ориентированный регламентирует первичность проектирования функциональных компонент по отношению к проектированию структур данных;</p> <ul style="list-style-type: none"> <li>- при подходе, ориентированном на данные, структуры данных определяются первыми, а процедурные компоненты являются производными от данных;</li> <li>- информационно-ориентированный подход позволяет работать с неиерархическими структурами данных</li> </ul>
по типу целевых систем	<ul style="list-style-type: none"> <li>- для систем реального времени (СРВ);</li> <li>- для информационных систем</li> </ul>	<p>Основная и первоочередная функция СРВ - реагирование на внешние события во времени</p>

В настоящее время успешно используются практически все известные методологии структурного анализа и проектирования, однако наибольшее распространение получили:

- SADT (Structured Analysis and Design Technique);
- методология структурного системного анализа Гейна-Сарсона (Gane-Sarson);
- методология структурного анализа и проектирования Йодана/Де Марко (Yourdon/De Marko);
- методология развития систем Джексона (Jackson);
- методология развития структурных систем Варнье-Орра (Warnier-Orr);
- методология анализа и проектирования систем реального времени Уорда-Меллора (Ward-Mellor) и Хатли (Hatley);
- методология информационного моделирования Мартина (Martin).[1]

В структурном анализе и проектировании применяются различные модели, которые описывают:

- 1) функциональную структуру проектируемой системы;
- 2) последовательность выполняемых процессов;
- 3) передачу информации между функциональными процессами (работами);
- 4) отношения между данными.

Наиболее распространенными моделями первых трех групп являются:

- функциональная модель SADT (Structured Analysis and Design Technique);
- модель IDEF3;
- DFD (Data Flow Diagrams) - диаграммы потоков данных.[5]

Модель «сущность — связь» (ERM — Entity-Relationship Model), описывающая отношения между данными, традиционно используется в структурном анализе и проектировании, однако, по существу, представляет собой подмножество объектной модели предметной области.[21]

В качестве инструментальных средств структурного анализа и проектирования выступают следующие диаграммы:

- BFD (Business Function Diagram) – диаграмма бизнес-функций (функциональные спецификации);
- DFD (Data Flow Diagram) – диаграмма потоков данных;
- STD (State Transition Diagram) – диаграмма переходов состояний (матрицы перекрестных ссылок);
- ERD (Entity Relationship Diagram) – ER-модель данных предметной области (информационно - логические модели “сущность - связь”);
- SSD (System Structure Diagram) – диаграмма структуры программного приложения.[4]

Диаграммы функциональных спецификаций (BFD) позволяют представить общую структуру ИС, отражающую взаимосвязь различных задач (процедур) в процессе получения требуемых результатов. Диаграммы этого вида определяют основные объекты и их графические образы в следующих нотациях: Йодана (Yourdon), Гейна – Сарсона (Gane – Sarson), SADT (Structured Analysis and Design Technique), SAG (Software AG).[10]

Следующим этапом развития графического языка описания функциональных систем SADT (Structured Analysis and Design Technique) является методология IDEF0. Целью методики является построение функциональной схемы исследуемой системы, описывающей все необходимые процессы с точностью, достаточной для однозначного моделирования деятельности системы. Обязательное наличие управляющих интерфейсных дуг является одним из главных отличий стандарта IDEF0 от других методологий классов DFD (Data Flow Diagram) и WFD (Work Flow Diagram).[7]

В следующей главе курсовой работы рассмотрим наиболее популярные методологии структурного анализа и проектирования информационных систем.

Вывод: в первой главе мы рассмотрели основные понятия проектирования информационных систем и методологию структурного подхода к проектированию ИС. Мы поняли, что информационная система - это упорядоченная совокупность документированной информации и информационных технологий.

Также мы выяснили, что сущность структурного подхода к проектированию информационных систем заключается в разбиении системы на функциональные подсистемы, которые, в свою очередь, делятся на подфункции, подразделяемые на задачи, и т.д. Процесс разбиения функций продолжается вплоть до конкретных процедур.

## **Глава 2. Анализ методологий структурного проектирования**

### **2.1 Метод функционального моделирования SADT (IDEF0)**

SADT (от англ. Structured Analysis and Design Technique) — методология структурного анализа и проектирования, объединяющая процесс моделирования, управления конфигурацией проекта, использование языковых средств и руководство проектом с собственным графическим языком.[20]

Методология SADT возникла в ходе революции конца 60-х годов и была вызвана сложностями в структурном программировании. В то время, когда большинство программистов билось над созданием программных средств, специалисты, традиционно занимавшиеся созданием крупномасштабных систем, стали осознавать необходимость большей упорядоченности. Таким образом, разработчики решили формализовать процесс создания системы, разбив его на фазы: анализ, проектирование, реализация, объединение, тестирование, установка — введение системы в действие, эксплуатация.[15]

Методология SADT, разработанная разработана Дугласом Россом в 1969-73гг., представляет собой совокупность методов, правил, процедур, предназначенных для построения функциональной модели некоторой предметной области. Функциональная модель SADT отображает функциональную структуру объекта, то есть производимые им действия и связи между ними.[9]

Основные элементы данной методологии основываются на концепциях:

1) графическое представление блочного моделирования объектов предметной области. Графика блоков и дуг SADT-диаграммы отображает в виде блока

функцию, а интерфейсы входа-выхода представляются дугами, входящими в блок и выходящими из него. Взаимодействие блоков друг с другом также описываются посредством интерфейсных дуг, которые определяют "ограничения", выражающие, когда и каким образом функции выполняются;

2) выполнение правил SADT требует строгости и точности, в то же время не накладывая чрезмерных ограничений на действия аналитика.[15]

Метод SADT считается классическим методом процессного подхода к управлению. Основным принцип процессного подхода заключается в структурировании деятельности организации в соответствии с ее бизнес-процессами, а не организационно-штатной структурой. В соответствии с этим принципом бизнес-модель должна выглядеть следующим образом:

1. Верхний уровень модели должен отражать контекст системы — взаимодействие моделируемого единственным контекстным процессом предприятия с внешним миром.
2. На втором уровне модели должны быть отражены основные виды деятельности (сгруппированные тематически бизнес-процессы) предприятия и их взаимосвязи. В случае большого их количества некоторые из них можно вынести на третий уровень модели. [17]

Дальнейшая детализация бизнес-процессов осуществляется посредством бизнес-функций — совокупностей операций, сгруппированных по определенным признакам. Описание элементарной бизнес-операции осуществляется заданием алгоритма ее выполнения.

В рамках технологии SADT было разработано несколько графических языков моделирования (таблица 2).

Таблица 2. Графические нотации методологии SADT

### **Нотация Назначение**

IDEF0	для документирования процессов производства и отображения информации об использовании ресурсов на каждом из этапов проектирования систем
-------	--

IDEF1	для документирования информации о производственном окружении систем
IDEF2	для отображения поведения систем во времени
IDEF3	для моделирования бизнес-процессов
IDEF4	объектно-ориентированное моделирование
IDEF5	моделирование наиболее общих (онтологических) закономерностей системы.

Метод SADT реализован в одном из стандартов этого семейства — IDEF. Семейство стандартов IDEF унаследовало свое обозначение от названия программы (IDEF=Icam DEFinition), последняя редакция была выпущена в 1993 г. Национальным Институтом по Стандартам и Технологиям (NIST).[6]

Методология IDEF0 является следующим этапом развития графического языка описания функциональных систем SADT. Исторически IDEF0 был разработан в 1981 году в качестве стандарта в рамках обширной программы автоматизации промышленных предприятий. Вместе со стандартом IDEF0 обычно используют стандарт моделирования процессов IDEF3 и моделирования данных IDEF1X.[9]

Целью методики IDEF0 является построение функциональной схемы исследуемой системы, описывающей необходимые процессы с точностью, достаточной для однозначного моделирования деятельности.[1]

В основе методологии лежат следующие понятия:

1) Функциональный блок (Activity Box) - представляет собой некоторую конкретную функцию в рамках исследуемой системы и изображается прямоугольником, каждая из его сторон имеет свое определенное значение (рис. 2):

Функция

A0

Управление (Control)

Выход (Output)

Вход (Input)

Механизм (Mechanism)

## **Рисунок 2. Функциональный блок и интерфейсные дуги**

2) Интерфейсная дуга (Arrow) - отображает элемент, который обрабатывается функциональным блоком системы или же оказывает другое влияние на функцию, представленную данным функциональным блоком. Интерфейсные дуги называют часто потоками или стрелками. С их помощью отображают различные объекты, определяющие в той или иной степени процессы, в системе: элементы реального мира (детали, сотрудники, товары) или потоки данных и информации (данные, документы, инструкции).

3) Декомпозиция (Decomposition) - ключевое понятие стандарта IDEF0. Принцип декомпозиции используется при разбиении сложного процесса системы на составляющие функции. Уровень детализации процесса при этом определяется непосредственно разработчиком модели ИС. Декомпозиция позволяет постепенно и структурировано представлять модель системы в виде иерархической структуры отдельных диаграмм.

4) Глоссарий (Glossary) подразумевает, что для каждого из элементов IDEF0 (диаграмм, интерфейсных дуг, функциональных блоков) подразумевается создание и поддержание набора соответствующих ключевых слов, определений, характеризующих объект, отображенный элементом.[3]

Модель (AS-IS, TO-BE или SHOULD-BE) может содержать **4 типа диаграмм**:

- контекстную диаграмму;
- диаграммы декомпозиции;
- диаграммы дерева узлов;
- диаграммы только для экспозиции (for exposition only, FEO).[19]

Построение модели начинается с представления всей системы в виде простейшего компонента (контекстной диаграммы) (рис.3).

A-0

A0

### **Рисунок 3. Функциональный блок и интерфейсные дуги IDEF0-модели**

Затем блок детализируется на другой диаграмме (декомпозиция) с помощью блоков, соединенных интерфейсными дугами (рис. 4).

A0

A1

A2

A3

A4

A4

A41

A42

A43

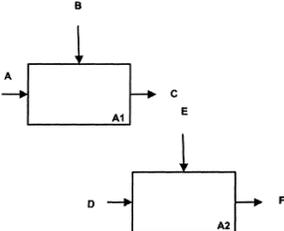
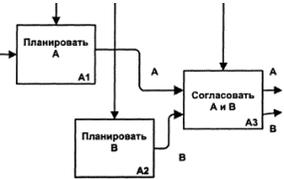
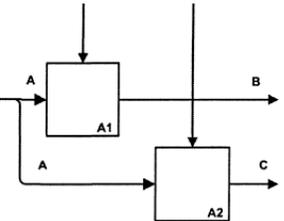
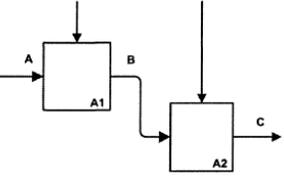
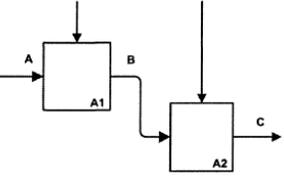
Верхняя диаграмма является родительской для нижней

### **Рисунок 4. Структура IDEF0-модели. Декомпозиция диаграмм**

Данные блоки определяют основные подфункции исходной функции. Такая декомпозиция выявляет полный набор подфункций, из которых каждая показана как блок, границы у которого определены интерфейсными дугами. Любая из таких подфункций может быть декомпозирована подобным образом в целях еще большей детализации.[2]

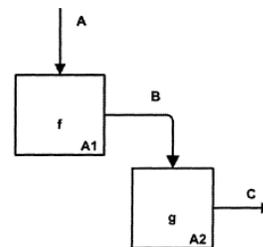
Одним из важных моментов при моделировании SADT (IDEF0) является точная согласованность типов связей между функциями. Различают следующие связи (в порядке возрастания их относительной значимости), которые описаны в таблице 3.

### **Таблица 3. Типы связей между функциями в методологии IDEF0**

Тип связи	Характеристика	Пример изображения на диаграмме
Случайная связь	показывает, что конкретная связь между функциями незначительна или полностью отсутствует	
Логическая связь	данные и функции собираются вместе благодаря тому, что они попадают в общий класс или набор элементов, но необходимых функциональных отношений между ними не обнаруживается	
Процедурная связь	функции сгруппированы вместе благодаря тому, что они выполняются в течение одной и той же части цикла или процесса	
Коммуникационная связь	функции группируются благодаря тому, что они используют одни и те же входные данные и/или производят одни и те же выходные данные	
Последовательная связь	выход одной функции служит входными данными для следующей. Связь между элементами на диаграмме является более тесной, чем в рассмотренных выше случаях, поскольку моделируются причинно-следственные зависимости	

Функциональная  
связь

все элементы функции влияют на выполнение одной и только одной функции. Диаграмма, являющаяся чисто функциональной, не содержит чужеродных элементов, относящихся к последовательному или более слабому типу связи



Пример построения диаграммы предметной области в нотации IDEF0 показан на рисунках приложения 1.

Методология IDEF0 может применяться для моделирования широкого круга информационных систем и определения требований и функций, а затем для разработки программного обеспечения, которое удовлетворяет данным требованиям и реализует такие функции.[13]

В настоящее время появляются десятки новых методологий моделирования деятельности предприятия и взглядов на её архитектуру, IDEF0 сохраняет актуальность для задач усовершенствования предприятий и организаций.

## 2.2 Метод моделирования процессов IDEF3

Метод моделирования IDEF3, являющийся частью семейства стандартов IDEF, был разработан в конце 1980-х г. для закрытого проекта ВВС США. Этот метод предназначен для моделей процессов, в которых важно понять последовательность выполнения действий и взаимозависимости между ними.

Хотя IDEF3 и не достиг статуса федерального стандарта США, он получил широкое распространение среди системных аналитиков в качестве дополнения к методу функционального моделирования IDEF0 (модели IDEF3 могут применяться для детализации функциональных блоков IDEF0, которые не имеют диаграмм декомпозиции).[6]

Основой модели нотации IDEF3 служит сценарий процесса, в котором выделяется последовательность действий и под процессами анализируемой системы.

Основной единицей модели IDEF3 является диаграмма. Другой ключевой компонент модели — действие или «единица работы» (Unit of Work - UOW) - отображается в виде прямоугольника.[12]

Действия на диаграмме именуются с использованием глаголов или отглагольных существительных, каждому из действий присваивается уникальный идентификационный номер, который не используется вновь даже в том случае, если действие в процессе построения модели удаляется. В диаграммах IDEF3 номер действия обычно предваряется номером его родителя (рис. 5).

## **Обработать заказ клиента**

### **Наименование**

### **действия**

### **Номер действия**

### **1.1**

## **Рисунок 5. Изображение и нумерация действия в диаграмме IDEF3**

Нотация IDEF3 использует следующие объекты (таблица 4).

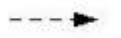
## **Таблица 4. Изображения объектов в методологии IDEF0**

<b>Наименование</b>	<b>Описание</b>	<b>Графическое представление</b>
Модель работы (UOW)	Объект служит для описания функций (процедур, работ), выполняемых подразделениями/ сотрудниками предприятия	
Ссылочный объект	Объект, используемый для описания ссылок на другие диаграммы модели, циклические переходы, различные комментарии к функциям	

Логическое «И»	Логический оператор, определяющий связи между функциями, позволяет описать ветвление	&
Логическое «ИЛИ»	Логический оператор, определяющий связи между функциями, позволяет описать ветвление	O
Логическое исключающее «ИЛИ»	Логический оператор, определяющий связи функциями в рамках процесса. Позволяет описать ветвление процесса	X

Существенные взаимоотношения между действиями изображаются с помощью связей. Возможные типы связей:

 - сплошная линия, связывающая единицы работ, показывает, что работа-источник должна закончиться прежде, чем начнется работа-цель.

 - пунктирная линия, для изображения связей между единицами работ и между единицами работ и объектами ссылок.

 - стрелка с двумя наконечниками, для описания того, что объект используется в двух или более единицах работы.

Все соединения на диаграммах должны быть парными, из чего следует, что любое разворачивающее соединение имеет парное себе сворачивающее. Однако типы соединений не должны обязательно совпадать. Соединения могут комбинироваться при создании более сложных ветвлений.[8]

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

Пример построения диаграммы предметной области в нотации IDEF3 показан на рисунках приложения 2.

IDEF3 фиксирует поведенческие аспекты реальных и проектируемых систем, позволяет описать временную последовательность происходящих событий, связанных причинно-следственными зависимостями. В отличие от языков и систем имитационного моделирования (например, SIMAN, SLAM, GPSS, WITNESS, Dynamo, Pilgrim), позволяющих реализовать математические модели процессов, IDEF3 ограничивается структурированным описанием объекта и может использоваться как подготовительная стадия разработки.[10]

Существует две разновидности описательных моделей IDEF3:

1) OSTD/OSTN (Object State Transition Description/Network) — описание (сеть) состояний и переходов;

2) PFD/PFDD (Process Flow Description/Diagram) — описание (диаграмма) потока процессов.[18]

## **2.3 Моделирование потоков данных**

Диаграммы потоков данных (Data Flow Diagrams — DFD) представляют собой иерархию функциональных процессов, связанных друг с другом потоками данных. Цель данного представления — продемонстрировать, каким образом каждый процесс преобразует входные данные в выходные и выявить отношения между этими процессами.[22]

Диаграммы потоков данных являются основным средством моделирования функциональных требований к проектируемой системе.

Для построения диаграммы DFD используются традиционно две различные нотации, соответствующие методам Йордона-ДеМарко и Гейна-Сэрсона. Эти нотации незначительно отличаются друг от друга графическим изображением символов. Рассмотрим особенности методологии DFD в нотации Гейна-Сэрсона.

При создании диаграммы потоков данных используются 4 основных понятия:

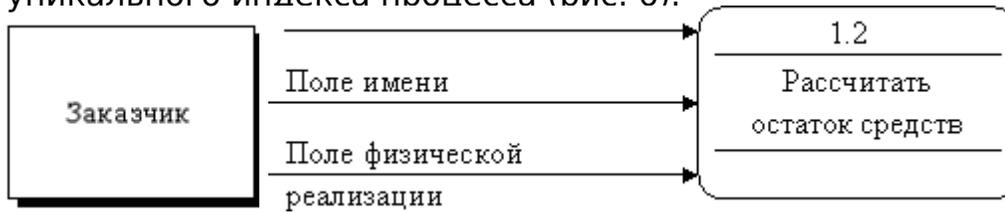
1) Потоки данных - являются абстракциями, применяемыми для моделирования передачи информации / физических компонент из одной части системы в другую.

2) Внешняя сущность - это материальный объект вне контекста информационной системы, являющийся источником/ приемником системных данных. Ее имя должно

содержать существительное, к примеру, «место хранения товаров».

Предполагается, что объекты, выделенные как внешние сущности, не должны участвовать ни в какой обработке.

3) Назначение процесса (работы) - продуцирование выходных потоков из входных в соответствии с действиями, задаваемыми именем процесса (имя должно содержать глагол в неопределенной форме с последующим дополнением, к примеру, «получить документы по реализации продукции»). Каждый процесс должен иметь уникальный номер для ссылок на него внутри диаграммы, который может использоваться совместно с номером диаграммы для получения в модели уникального индекса процесса (рис. 6).



а) внешняя сущность

б) процесс

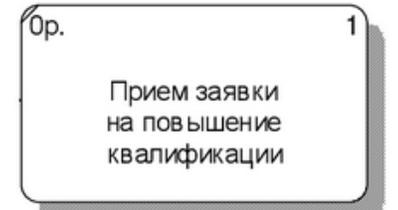
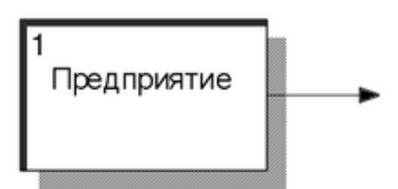
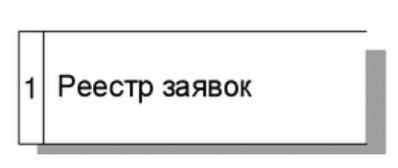
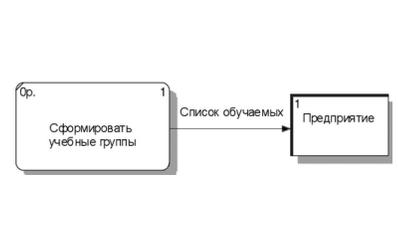
### Рисунок 6. Изображение основных понятий в DFD

4) Хранилище (или накопитель) данных позволяет на указанных участках модели определять данные, которые будут сохраняться между процессами в памяти. Хранилище фактически представляет собой «срезы» потоков данных во времени. Информация из хранилища может использоваться в любое время после получения, данные могут выбираться в любом порядке.[1]

Основные элементы диаграммы DFD в нотации Гейна-Сэрсона показаны в таблице 5.

**Таблица 5.** Изображения объектов в методологии DFD в нотации Гейна-Сэрсона

Наименование	Описание	Графическое представление
--------------	----------	---------------------------

Функциональный блок	Моделирует некоторую функцию, преобразующую вход в выход. Функциональные блоки изображаются в виде прямоугольников со скругленными углами	
Внешняя сущность	Обеспечивает необходимые входы и (или) выходы для функциональных блоков	
Хранилище данных	Механизм, который поддерживает хранение данных для их промежуточной обработки	
Потоки данных	Описывают перемещение данных между частями системы. Стрелки, потоков могут быть однонаправленными, двунаправленными, могут начинаться и заканчиваться на любой стороне блока	

В состав DFD также входят словари данных (каталоги всех элементов данных в DFD, включая групповые и индивидуальные потоки данных, хранилища и процессы и их атрибуты) и мини спецификации (описывают DFD-процессы нижнего уровня, то есть представляют собой алгоритмы описания задач, выполняемых процессами).[9]

Процесс построения DFD начинается с создания так называемой основной диаграммы типа «звезда», на которой представлен моделируемый процесс и все внешние сущности, с которыми он взаимодействует. В случае сложного основного процесса он сразу представляется в виде декомпозиции на ряд взаимодействующих процессов. Для всех внешних сущностей строится таблица событий, описывающая их взаимодействие с основным потоком. После декомпозиции основного процесса для каждого под процессом строится аналогичная таблица внутренних событий.

Пример построения диаграммы предметной области в нотации DFD показан на рисунках приложения 3.

## 2.4 Диаграммы переходов состояний (SDT)

Диаграммы переходов состояний моделируют поведение системы во времени в зависимости от происшедших событий.

SDT (State Transition Diagrams) – диаграммы переходов состояний, предназначена для моделирования аспектов функционирования системы, зависящих от времени или реакции на события.[16]

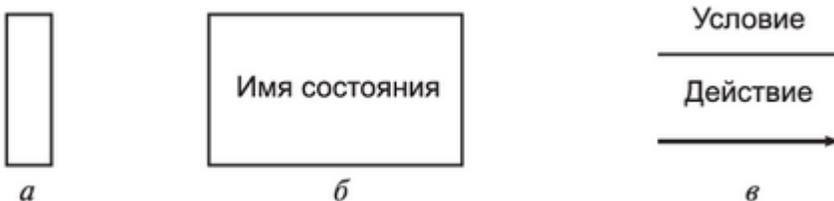
С помощью диаграмм перехода состояний можно моделировать последующее функционирование системы исходя из предыдущих и текущего состояний.

В SDT-диаграммах узлы соответствуют состояниям динамической системы, а дуги — переходу системы из одного состояния в другое. Узел, из которого выходит дуга, является начальным состоянием, узел, в который дуга входит, — следующим. Дуга помечается именем входного сигнала или события, вызывающего переход, а также сигналом или действием, сопровождающим переход.[22]

Применяются 2 способа построения модели STD. Первый заключается в идентификации всех возможных состояний и дальнейшем исследовании всех не бессмысленных связей (то есть переходов) между ними. По второму способу строится сначала начальное состояние, затем следующее за ним и т.д.

Основными элементами данной модели служат понятия начальное (терминальное) состояние, переход, условие и действие, описывающие поведение системы во времени в контексте наступающих событий.[19]

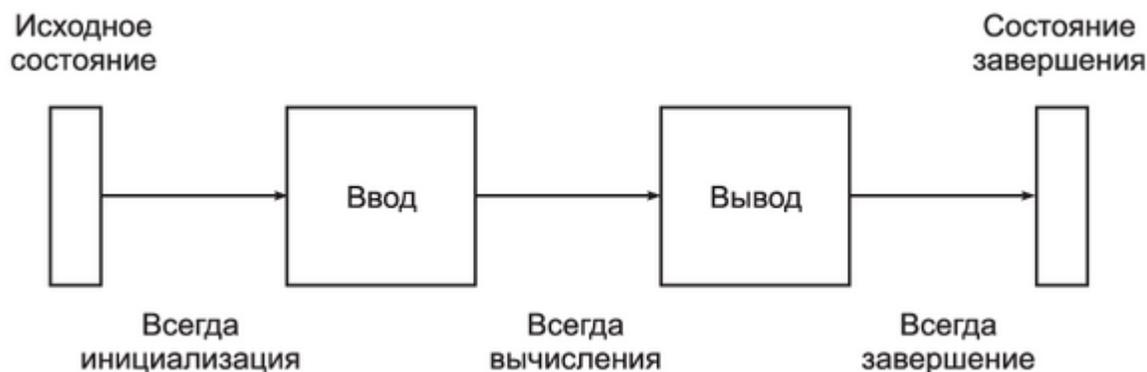
На рисунке 7 представлены условные обозначения в SDT-диаграммах.



**Рисунок 7. Условные обозначения диаграмм переходов состояний:**

а – терминальное состояние; б – промежуточное; в - переход

На рисунке 8 представлена диаграмма переходов состояний для программы, активно не взаимодействующей с окружающей средой, имеющей простой интерфейс, которая производит некоторые вычисления и выводит простой результат.



### Рисунок 8. Пример SDT-диаграммы

Диаграммы переходов состояний обычно используются для описания отношения между входными и выходными управляющими потоками на управляющем процессе-предке и позволяют осуществлять декомпозицию управляющих процессов.

Вывод: Во второй главе мы рассмотрели такие методологии, как:

1. SADT — это методология структурного анализа и проектирования, которая объединяющая процесс моделирования, управления конфигурацией проекта, использование языковых средств и руководство проектом с собственным графическим языком.

Он считается классическим методом процессного подхода к управлению. Основной принцип процессного подхода заключается в структурировании деятельности организации в соответствии с ее бизнес-процессами, а не организационно-штатной структурой.

1. Метод моделирования IDEF3, являющийся частью семейства стандартов IDEF, был разработан в конце 1980-х г. для закрытого проекта ВВС США. Этот метод предназначен для моделей процессов, в которых важно понять последовательность выполнения действий и взаимозависимости между ними.

2. Диаграммы потоков данных (Data Flow Diagrams — DFD) представляют собой иерархию функциональных процессов, связанных друг с другом потоками данных. Цель данного представления — продемонстрировать, каким образом каждый процесс преобразует входные данные в выходные и выявить отношения между этими процессами.

Они являются основным средством моделирования функциональных требований к проектируемой системе. Диаграммы переходов состояний моделируют поведение системы во времени в зависимости от происшедших событий.

1. SDT – диаграммы переходов состояний, предназначена для моделирования аспектов функционирования системы, зависящих от времени или реакции на события. С помощью нее можно моделировать последующее функционирование системы исходя из предыдущих и текущего состояний.

## **Глава 3. Сравнительный анализ и оценка средств реализации структурных методов анализа и проектирования информационных систем**

В настоящее время известно около 90 разновидностей методологий структурного системного анализа, которые можно классифицировать по отношению к школам (для моделирования программных систем или вообще систем), по порядку построения модели (декларирующие первичность функционального или информационного моделирования), по типу целевых систем (информационные системы или системы реального времени). Несмотря на такое обилие методов, практически во всех используются 3 группы средств:

- 1) DFD (Data Flow Diagrams) - диаграммы потоков данных или SADT-диаграммы, иллюстрирующие функции, которые должна выполнять система;
- 2) ERD (Entity-Relationship Diagrams) - диаграммы "сущность-связь", моделирующие отношения между данными;
- 3) STD (State Transition Diagrams) - диаграммы переходов состояний, моделирующие зависящее от времени поведение системы (аспекты реального времени).[11]

Так как в настоящее время практически нет альтернативы методологиям ERD и STD, используемых для, соответственно, информационного и поведенческого

моделирования, интерес представляет сравнительный анализ средств функционального моделирования, а именно, DFD и SADT-диаграмм.

С этой точки зрения все разновидности структурного системного анализа могут быть разбиты на две группы - применяющие методы и технологию DFD (в различных нотациях) и использующие SADT-методологию.

Сравнительный анализ данных методов структурного анализа проводится по следующим параметрам.

1. Адекватность средств решаемым задачам. Модели SADT (IDEF0) используются традиционно для моделирования организационных систем (бизнес-процессов). С другой стороны, не существует принципиальных ограничений на использовании модели DFD в качестве средства моделирования бизнес-процессов.

Практически любой класс систем успешно моделируется при помощи DFD-ориентированных методов, а SADT-диаграммы оказываются значительно менее выразительными и удобными при моделировании ПО.

2. Выразительные средства. В SADT отсутствуют вообще выразительные средства для моделирования особенностей ИС, а DFD с самого начала создавались как средства проектирования ИС (тогда как SADT — как средство моделирования систем вообще) и имеют более богатый набор элементов, более адекватно отражающих специфику таких систем (к примеру, хранилища данных являются прообразами файлов или баз данных, внешние сущности отражают взаимодействие моделируемой системы с внешним миром).

В модели SADT жестко типизированы дуги (вход, выход, управление, механизм), однако применительно к программам смысловое различие между входами и выходами стирается. При этом анализ системы при помощи потоков данных и процессов, их преобразующих, является более прозрачным и недвусмысленным.

3. Согласованность с другими средствами структурного анализа. Так как SADT-диаграммы предназначены для моделирования систем общего класса, в них отсутствуют средства описания данных и событий, то согласование модели, к примеру, с моделями ERD- или STD практически невозможно.

В свою очередь, модели DFD-, ERD- и STD- дополняют взаимно друг друга и по сути являются согласованными представлениями различных аспектов одной и той же модели.

Интеграция DFD→STD может быть осуществлена за счет расширения классической DFD специальными средствами проектирования систем реального времени (потоками, управляющими процессами, хранилищами данных). Интеграция DFD→ERD осуществляется с использованием отсутствующего объекта в SADT - хранилища данных, структура которого описывается в модели ERD и согласуется в DFD по соответствующим потокам и другим хранилищам.

4. Интеграция с последующими этапами. Важная характеристика любой методологии - ее совместимость с последующими этапами применения результатов анализа. DFD могут быть легко преобразованы в модели проектирования (структурные карты) - это близкие модели.

5. Ограничения модели. Жесткие ограничения SADT, запрещающие использовать более 6—7 блоков на диаграмме, в ряде случаев вынуждают искусственно детализировать процесс, что затрудняет понимание модели заказчиком, резко увеличивает ее объем и, как следствие. В качестве примера можно рассмотреть модель операций по снятию денег с вклада физического лица в банке. Для моделирования этих операций целесообразно использовать единственную диаграмму DFD, так как все операции без исключения имеют одни и те же входы (сберегательная книжка и расходный ордер) и выходы (сберегательная книжка и наличные деньги), различаются лишь механизмами начисления процентов. Если попытаться структурировать данные операции в соответствии с ограничениями SADT путем группирования по какому-либо признаку (срочные, пенсионные и т.п.), то получится минимум 6 диаграмм, сложность каждой из которых не меньше сложности единственной DFD диаграммы.

Таким образом, метод SADT успешно работает только при описании хорошо специфицированных и стандартизованных бизнес-процессов. К примеру, он четко работает в зарубежных корпорациях, поэтому этот метод и принят в США в качестве типового.

В большинстве российских организаций бизнес-процессы начали формироваться и развиваться сравнительно недавно, они слабо типизированы, поэтому разумнее ориентироваться на модели, основанные на потоковых диаграммах.

DFD-диаграммы можно применять при проектировании любых программных систем, тогда как SADT-диаграммы лучше применять при моделировании систем - бизнес-процессов предприятий.

Рассмотрим достоинства и недостатки рассмотренных средств реализации структурных методов анализа и проектирования информационных систем в таблице 6.

Таблица 6. Сравнительная характеристика структурных методологий

Название методики	Достоинства	Недостатки	Сфера применения
IDEFO	<ul style="list-style-type: none"> <li>- полнота описания бизнес-процесса (управление, информационные и материальные потоки, обратные связи);</li> <li>- комплексность декомпозиции;</li> <li>- возможность агрегирования и детализации потоков данных и информации (разделение и слияние дуг);</li> <li>- наличие жестких требований, обеспечивающих получение моделей стандартного вида;</li> <li>- простота документирования</li> </ul>	<ul style="list-style-type: none"> <li>- сложность восприятия (большое число дуг на диаграммах);</li> <li>- большое число уровней декомпозиции;</li> <li>- трудность увязки нескольких процессов, представленных в различных моделях одной и той же организации</li> </ul>	<p>В наибольшей степени подходит для описания процессов верхнего уровня управления, для моделирования бизнес-процессов организации</p>

- позволяет точно  
показать порядок  
действий,  
разветвления и  
параллельные  
процессы;

IDEF3

- не ограничивает  
аналитика чрезмерно  
жесткими рамками  
синтаксиса, что может  
привести к созданию  
неполных или противो-  
речивых моделей;

- содержит все  
необходимое для  
построения моделей,  
которые в дальнейшем  
могут быть  
использованы для  
имитационного  
анализа

- отсутствие переноса  
и контроля стрелок на  
детализацию;

- отсутствие  
возможности  
расщепления и  
слияния модели

Диаграммы DDEF3  
удобно использовать,  
где нет документов  
(например, в описании  
действий автомата),  
IDEF3 может быть  
использован в  
дополнении к  
диаграмме IDEF0

	- возможность однозначно определить внешние сущности, анализируя потоки информации внутри и вне системы;	- необходимость искусственного ввода управляющих процессов, поскольку управляющие воздействия (потоки) и управляющие процессы в DFD ничем не отличаются от обычных;	
DFD	- возможность проектирования сверху вниз, что облегчает построение «как должно быть»;		Используется для описания документооборота и обработки информации. DFD-диаграмма легко преобразуется в UML-документ
	- наличие спецификаций процессов нижнего уровня, что позволяет преодолеть незавершенность функциональной модели	- отсутствие понятия времени (анализа временных промежутков)	

С точки зрения бизнес-моделирования информационных систем каждый из представленных структурных подходов обладает своими преимуществами. Функциональное моделирование на основе SADT хорошо показывает себя в тех случаях, когда организационная структура находится в процессе изменения или вообще слабо оформлена. Подход от выполняемых функций интуитивно лучше понимается исполнителями при получении от них информации об их текущей работе. При этом функциональная модель IDEF0 отображает производимые действия над объектом и связи между ними, IDEF3 – методика моделирования процессов служит для создания таких моделей, в которых важно понять последовательность выполнения действий и взаимозависимостей между ними. Средства построения моделей SADT целесообразно использовать на ранних этапах жизненного цикла ИС для понимания системы до ее воплощения. SADT позволяет сократить дорогостоящие ошибки на ранних этапах создания системы, улучшить контакт между пользователями и разработчиками, сгладить переход от анализа к проектированию.

Моделирование на основе диаграмм DFD следует применять к системам обработки информации, а не к системам вообще, как это предполагается в SADT. Практически любой класс систем успешно моделируется при помощи DFD-ориентированных методов: в этом случае вместо реальных объектов рассматриваются отношения, описывающие свойства этих объектов и правила их поведения. Примерами таких систем служат системы документооборота, управления и другие системы, богатые разнообразными отношениями.

Вывод: Сравнительный анализ и оценка средств реализации структурных методов анализа и проектирования информационных систем в главе три показал нам, что каждый из представленных структурных подходов обладает своими преимуществами и недостатками. К примеру, IDEF0 сложно воспринимается, в нем много уровней декомпозиции, но при этом полностью описывает бизнес-процессы, дает возможность агрегирования и детализации потоков данных и информации. И таким образом в наибольшей степени подходит для описания процессов верхнего уровня управления, для моделирования бизнес-процессов организации

## **ЗАКЛЮЧЕНИЕ**

В основе проектирования информационных систем лежит моделирование предметной области, позволяющее уменьшить время и сроки проведения проектировочных работ и получить более эффективный и качественный проект.

Процесс бизнес-моделирования предметной области может быть реализован в рамках различных методик, которые отличаются прежде всего, своим подходом к тому, что представляет собой моделируемая организация. В соответствии с различными представлениями об организации методики принято делить на объектные и структурные (или функциональные).

Структурное моделирование хорошо показывает себя в тех случаях, когда организационная структура находится в процессе изменения или вообще слабо оформлена. Подход от выполняемых функций интуитивно лучше понимается исполнителями при получении от них информации об их текущей работе.

Структурное проектирование основано на детализации метода нисходящего проектирования. Процесс декомпозиции в структурном проектировании реализован в потоках данных в системе.

В настоящее время известно около 90 разновидностей методологий структурного системного анализа, которые можно классифицировать по различным признакам. Несмотря на такое обилие методов, практически во всех используются три группы средств:

- 1) DFD (Data Flow Diagrams) - диаграммы потоков данных или SADT-диаграммы, иллюстрирующие функции, которые должна выполнять система;
- 2) ERD (Entity-Relationship Diagrams) - диаграммы "сущность-связь", моделирующие отношения между данными;
- 3) STD (State Transition Diagrams) - диаграммы переходов состояний, моделирующие зависящее от времени поведение системы (аспекты реального времени).

Наибольшее распространение для целей функционального моделирования получили графические модели на основе SADT - модель IDEF0, модель IDEF3, расширяющая возможности IDEF0, DFD - диаграммы потоков данных.

С точки зрения бизнес-моделирования информационных систем каждый из представленных структурных подходов обладает своими преимуществами. Функциональная модель IDEF0 отображает производимые действия над объектом и связи между ними, IDEF3 – методика моделирования процессов служит для создания таких моделей, в которых важно понять последовательность выполнения действий и взаимозависимостей между ними. DFD следует применять к программным системам обработки информации, а не к системам вообще, как это предполагается в SADT. С помощью DFD-ориентированных методов рассматриваются отношения, описывающие свойства объектов системы и правила их поведения.

## **СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ**

1. Абдикеев, Н.М. Корпоративные информационные системы управления: Учебник / Под науч. ред. д-ра техн. наук, проф. Н. М. Абдикеева, канд. физ.-мат. наук, доп. О.В. Китовой. - М: ИНФРА-М, 2011. - 464 с.
2. Вдовин, В. М. Теория систем и системный анализ [Электронный ресурс] : Учебник для бакалавров / В. М. Вдовин, Л. Е. Суркова, В. А. Валентинов. - 3-е изд. - М.: Издательско-торговая корпорация «Дашков и К°», 2013. - 644 с.

3. Вендров А.М. Проектирование программного обеспечения экономических информационных систем: Учебник. - 2-е изд., перераб. и доп. — М.: Финансы и статистика, 2008. — 544 с.
4. Гагарина Л. Г. Технология разработки программного обеспечения: Учеб. пос. / Л.Г.Гагарина, Е.В.Кокорева, Б.Д.Виснадул; Под ред. проф. Л.Г.Гагариной - М.: ИД ФОРУМ: НИЦ Инфра-М, 2013. - 400 с.
5. Гвоздева В. А. Основы построения автоматизированных информационных систем: Учебник / В.А. Гвоздева, И.Ю. Лаврентьева. - М.: ИД ФОРУМ: НИЦ Инфра-М, 2013. - 320 с.
6. Душин, В. К. Теоретические основы информационных процессов и систем [Электронный ресурс] : Учебник / В. К. Душин. - 5-е изд. - М.: Издательско-торговая корпорация «Дашков и К°», 2014
7. Елиферов, В. Г. Бизнес-процессы: Регламентация и управление: Учебник / В.Г. Елиферов, В.В. Репин; Институт экономики и финансов "Университет". - М.: НИЦ ИНФРА-М, 2013. - 319 с.
8. Затонский А. В. Информационные технологии: разработка информационных моделей и систем: Учеб. пос. / А.В.Затонский - М.: ИЦ РИОР: НИЦ ИНФРА-М, 2014 – 344 с.
9. Информационная система предприятия: Учеб. пособие / Л.А. Вдовенко. - М.: Вузовский учебник: ИНФРА-М, 2010. - 237 с.
10. Исаев Г. Н. Моделирование информационных ресурсов: теория и решение задач: учебное пособие / Г.Н. Исаев. - М.: Альфа-М: ИНФРА-М, 2010. - 224 с.
11. Калянов Г. Н. CASE-технологии. Консалтинг в автоматизации бизнес-процессов. – 3-е изд. – М.: Горячая линия-Телеком, 2002. -320 с.
12. Кирюков С.Р. К вопросу о структурном моделировании в управлении. – Самара: СИУ, 2009. – 180 с.
13. Макарова, Н. В. Информатика: Учебник для вузов / Н. В. Макарова, В. Б. Волков — СПб.: Питер, 2011. — 576 с.
14. Маклаков С. Моделирование бизнес-процессов с AllFusion Process Modeler. – М.: Диалог-МИФИ, 2009. - 240 с.
15. Назаров С. В. Архитектура и проектирование программных систем: Монография / С.В. Назаров. - М.: НИЦ Инфра-М, 2013. - 351 с.
16. Пирогов, В. Ю. Информационные системы и базы данных: организация и проектирование: учеб. пособие / В. Ю. Пирогов. — СПб.: БХВ-Петербург, 2009. — 528 с.
17. Проектирование информационных систем: Учебное пособие / Н.З. Емельянова, Т.Л. Партыка, И.И. Попов. - М.: Форум: НИЦ ИНФРА-М, 2014. - 432 с.

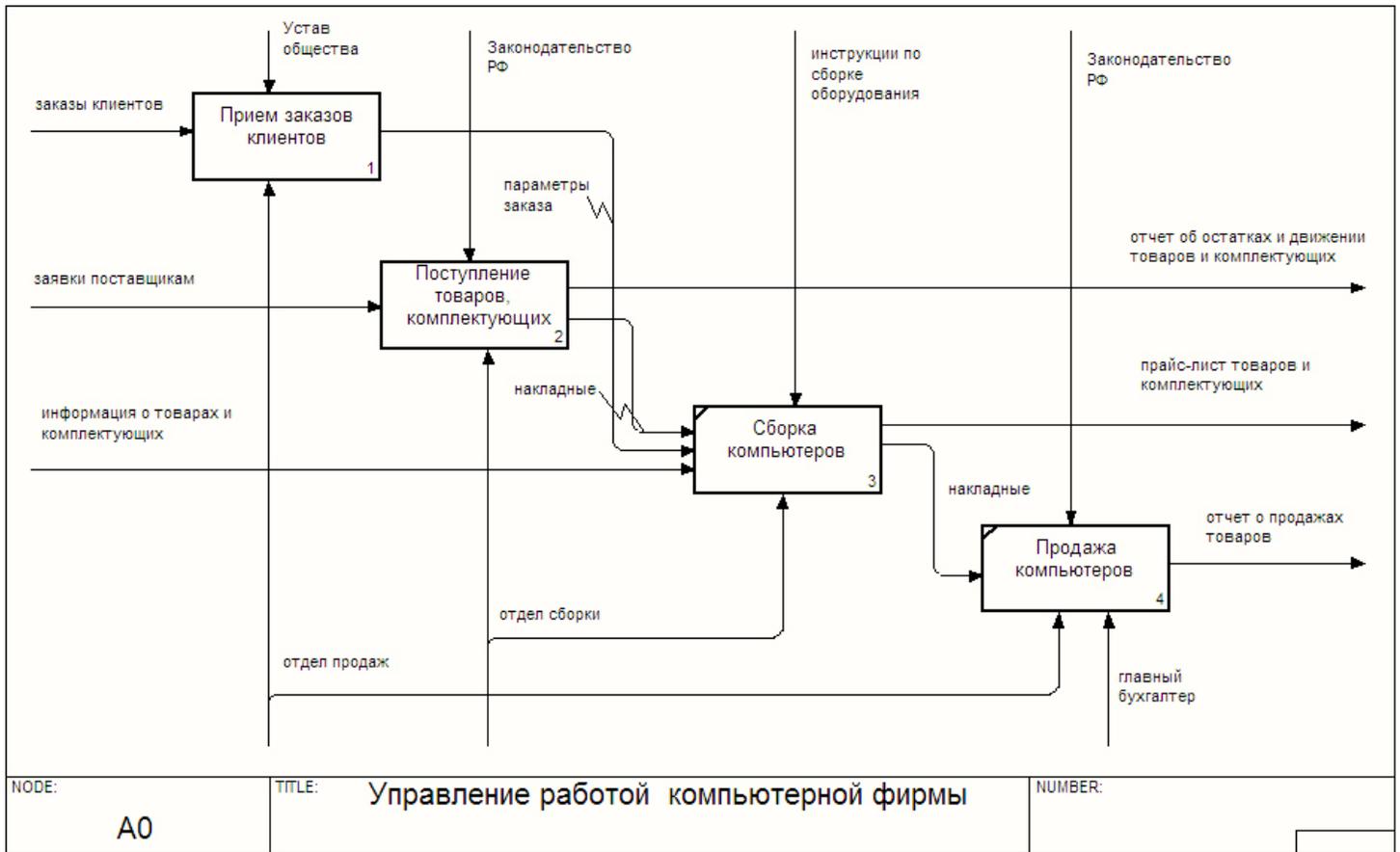
18. Проектирование информационных систем: Учебное пособие / Н.Н. Заботина. - М.: НИЦ ИНФРА-М, 2014. - 331 с.
19. Сравнение и проблема выбора методов структурного системного анализа [Электронный ресурс]. - Режим доступа: <http://www.pcweek.ru/themes/detail.php?ID=75209>
20. Сравнительный анализ структурных методологий [Электронный ресурс]. - Режим доступа: <http://www.osp.ru/dbms/1997/05-06/13031567/>
21. Титоренко, Г.А. Информационные системы и технологии управления: Учебник /Под ред. проф. Г.А. Титоренко. — М: ЮНИТИ-ДАНА, 2012. - 591 с.
22. Черников, Б. В. Информационные технологии управления: Учебник / Б.В. Черников. - 2-е изд., перераб, и доп. - М.: ИД ФОРУМ: НИЦ Инфра-М, 2013. - 368 с.

## ПРИЛОЖЕНИЕ 1

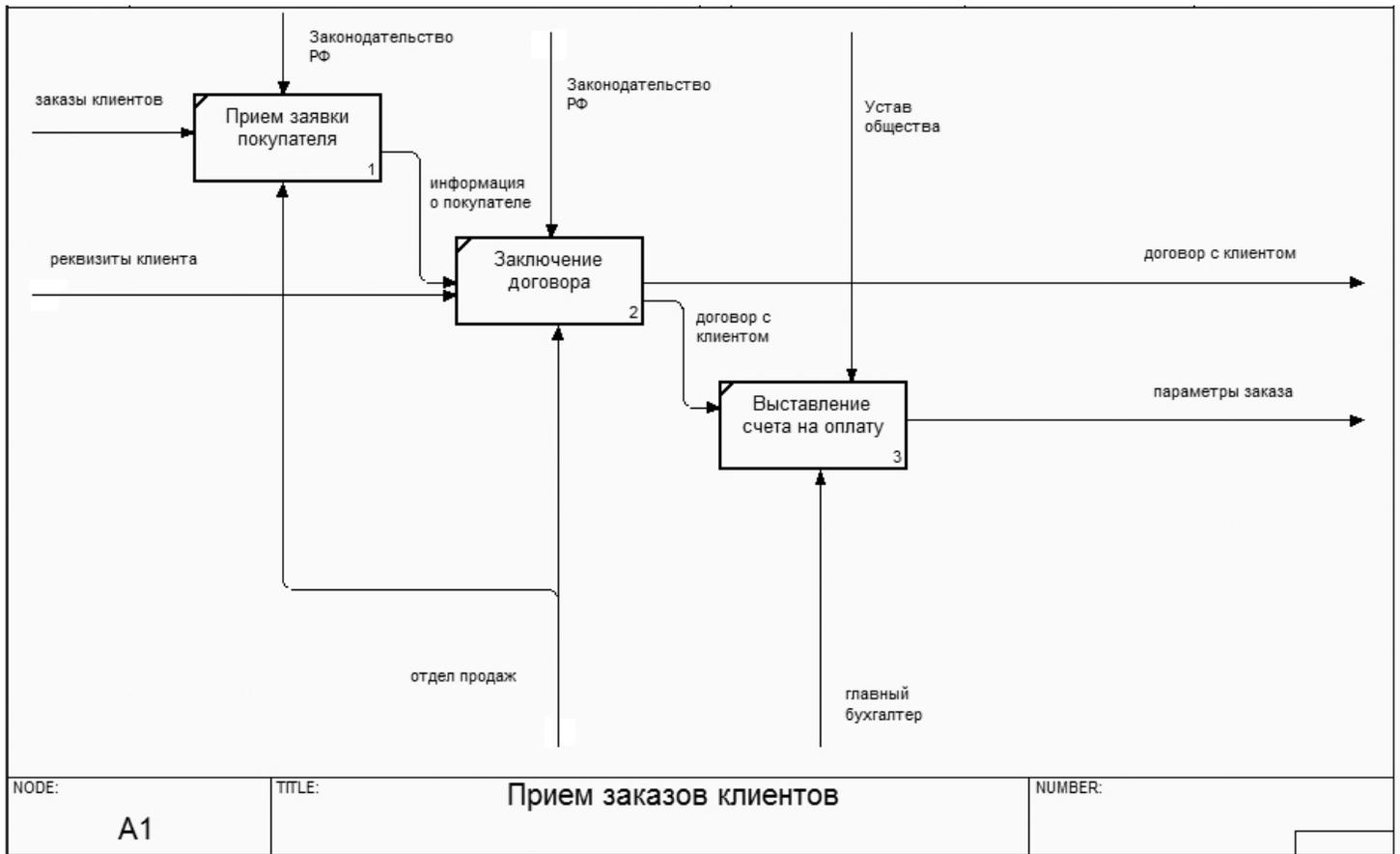
Пример построения IDEF0-модели



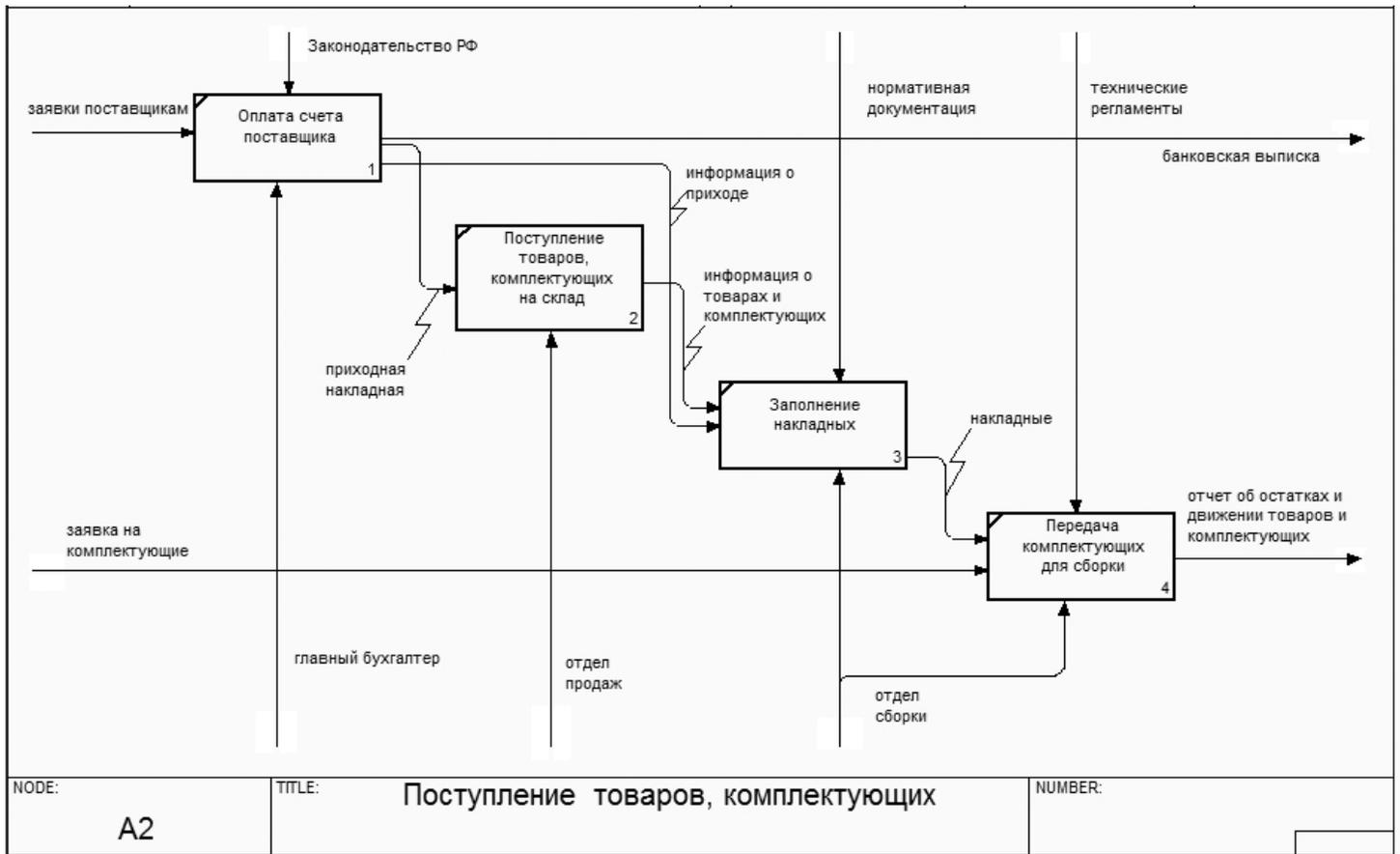
**Рисунок 1. Контекстная диаграмма**



**Рис.2 Декомпозиция контекстной диаграммы**



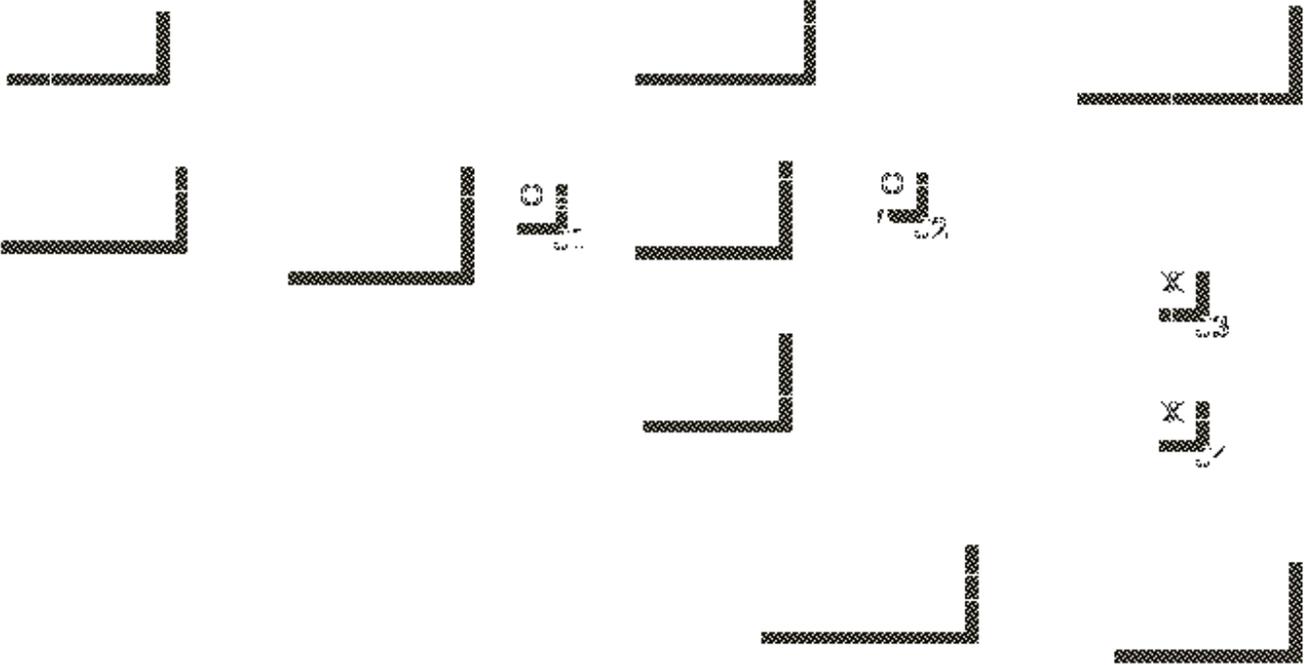
**Рис.3** Декомпозиция функционального блока



**Рис.4 Декомпозиция функционального блока**

ПРИЛОЖЕНИЕ 2

Пример построения IDEF3-диаграммы



### ПРИЛОЖЕНИЕ 3

Пример построения DFD-диаграммы

