Глава 6. Подходы к проектированию ИС.6.1 Анализ и проектирование ИС.

Анализ требований – процесс жизненного цикла программы, во время которого требования уточняются, формализуются и документируются.

Проектирование – процесс жизненного цикла программы, во время которого исследуется ее структура и взаимосвязи элементов. Состоит из уровней проектирования архитектуры и детального проектирования.

Существует два типа методологий – структурная и объектно-ориентированная.

6.1.1 Методы проектирования архитектур ИС.

Проектирование архитектуры:

Структурная методология.

- *Метод нисходящего проектирования* представлен двумя стратегиями:

- пошаговое уточнение, на каждом следующем этапе декомпозиции определяются подсистемы более низкого уровня.

- анализ сообщений, анализируются потоки данных, обрабатываемые подсистемами.

- *Метод восходящего* проектирования – в первую очередь определяются вспомогательные подсистем.
- *Метод расширения ядра* – основное внимание уделяется выявлению множества подсистем, а не определению функций всей системы в целом.

Объектно-ориентированная методология.

- *Метод проектирования предметных областей* – выделение предметной области системы с точки зрения пользователя.

- *Метод наведения мостов* – одна предметная область использует механизмы и возможности, обеспечиваемые другой предметной областью. Мост между двумя предметными областями представляет собой набор предложений (с точки зрения пользователя) и набор требований (с точки зрения исполнителя).

Проектирование подсистем:

Структурная методология включает следующие основные методы проектирования модулей:

- - диаграммы «сущьность-связь».
- - структурные карты.
- - диаграммы деятельности.
- - диаграммы Варнье-Орра.
- - диаграммы переходов состояний.
- - блок-схемы.
- - псевдокод.

Объектно-ориентированная методология включает такие методы:

- - диаграммы кооперации.
- - диаграммы компонентов.
- - диаграммы развертывания.

Методы анализа и построения спецификаций:

Структурная методология включает следующие методы:

- - диаграммы потоков данных.
- - диаграммы потоков управления.
- - таблицы решений.
- - сети Петри.
- - диаграммы зависимости.
- - диаграммы декомпозиции.
- - диаграммы функционального моделирования.

Объектно-ориентированная методология включает такие методы:

- - КОК-карты (класс – ответственность – кооперация).
- - диаграммы вариантов использования.
- - диаграммы классов.
- - диаграммы состояний.
- - диаграммы деятельности.
- - диаграммы последовательности.

6.1.2 Подходы к ведению анализа и проектирования.

Структурная методология.

Три подхода на основе порядка построения модели:

1) *Процедурно-ориентированные подходы* – первично проектирование функциональных компонентов.
2) *Подходы, ориентированные на данные* – первичны входные и выходные данные, а функциональные компоненты вторичны.
3) *Информационно-ориентированные подходы* – отличается от предыдущей, что работа ведется с неиерархическими структурами данных.

Выделяют два класса целевых систем:

- *Информационные системы* (управляемые данными) – работают с большим объемом входных данных сложной структуры.
- *Системы реального времени* (управляемые событиями) – работаю с малым количеством входных данных простой структуры.

Группы средств моделирования:

- диаграммы, иллюстрирующие функции, которые система должна выполнять, и связи между ними, например, диаграммы потоков данных и функционального моделирования.
- диаграммы, моделирующие данные и их отношения, например, диаграммы «сущьность-связь».

Объектно-ориентированная методология.

Основные подходы:

1) *Подход на основе языка UML*. Состоит из четырех основных фаз, причем работа с диаграммами ведется в основном на второй и третьей фазах. На второй фазе, исследования, создается модель предметной области. На третьей фазе, построения, продолжается итеративная работа с диаграммами классов и деятельности. К ним добавляются типы диаграмм, которые определяют взаимодействие: диаграммы последовательности, диаграммы кооперации. В случае сложного поведения системы разрабатывается еще одна группа диаграмм – диаграммы состояний.

1) *Подход Шлеер-Меллора* использует три группы средств для создания модели предметной области:

- *информационное моделирование* – для определения отношений между данными (диаграмма классов).
- *моделирование состояний* – для определения зависящего от времени поведения системы (диаграмма состояний).
- *моделирование процессов* – для определения функций которая система должна выполнять (диаграмма деятельности, диаграмма последовательности).

Для анализа больших предметных областей используются диаграммы, по смыслу близкие к следующим диаграммам языка UML: диаграммы кооперации, компонентов, развертывания.

Подход предлагает механизм поддержки моделей состояний. Для этого вводятся четыре архитектурных класса:

- переход, описывающий каждый переход для всех моделей состояний в программе.
- конечная модель состояний, связывающая все экземпляры перехода, которые составляют одну модель состояний.
- активный экземпляр. Это абстрактный класс, из которого все экземпляры, имеющие модель состояний, наследуют их текущее состояние.
- таймер, обеспечивающий механизм работы таймеров на основе аппаратных средств, доступных для хранения следа времени.

1) *Подход Град и Буча.*
2) *Подход Джеймса Рамбо.*
3) *Подход Ивара Якобсона.*

6.2 Структурный подход к проектированию ИС.

Сущность структурного подхода к разработке ИС в ее декомпозиции (разбиении) на автоматизированные функции: система разбивается на функциональные подсистемы, которые, в свою очередь, делятся на подфункции, подразделяемые на задачи и т.д. Разбиение продолжается вплоть до конкретных процедур. При этом система сохраняет целостное представление.

6.2.1 Структурный анализ в проектировании ИС.

Проблемы, с которыми сталкивается системный аналитик а так же основные принципы анализа систем рассмотрены в первой главе.

В структурном анализе применяются в основном две группы средств, отражающие функции, выполняемые системой, и отношения между ними. Каждой группе средств соответствуют определенные виды моделей (диаграмм), самые распространенные:

- SADT-модели и соответствующие функциональные диаграммы.
- DFD-диаграммы потоков данных.
- ERD-диаграммы «сущьность-связь».

6.2.2 Классификация структурных методологий.

Современные структурные методологии анализа и проектирования классифицируются по следующим признакам:

- Отношению к школам – Software Engineering (SE) и Information Engineering (IE).
- Порядку построения модели – процедурно-ориентированные, ориентированные на данные и информационно-ориентированные.
- Типу целевых систем – для систем реального времени и для информационных систем.

Отличие информационных систем от систем реального времени:

Информационные системы	Системы реального времени
Управляемые данными	Управляемые событиями
Сложные структуры данных	Простые структуры данных
Большой объем входных данных	Малое количество входных данных
Интенсивный ввод\вывод	Интенсивные вычисления
Машинная независимость	Машинная зависимость

Классификация применяемых методологий:

Методология	Частота	Признаки классификации	
Йодан Де Марко	36,5%	SE	Процедурно-ориентированная ИС, СРВ
Гейн-Сарсон	20,2%	SE	Процедурно-ориентированная ИС, СРВ
Константайн	10,6%	SE	Процедурно-ориентированная ИС, СРВ
Джексон	7,7%	SE	Ориентированная на данные ИС, СРВ
Варнье-Орр	5,8%	SE	Ориентированная на данные ИС
Мартин	22,1%	IE	Информационно-ориентированная ИС
SADT	3,3%	IE	Варианты использования ИС: - процедурно-ориентированная. - ориентированная на данные.
Stradis	1,9%	IE	Процедурно-ориентированная ИС

6.2.3 Методология функционального моделирования.

Методология SADT разработана Дугласом. На ее основе разработана методология IDEF0, которая является основной частью программы ICAM (Интеграция компьютерных и промышленных технологий).

Основные элементы концепции основаны на следующих концепциях:

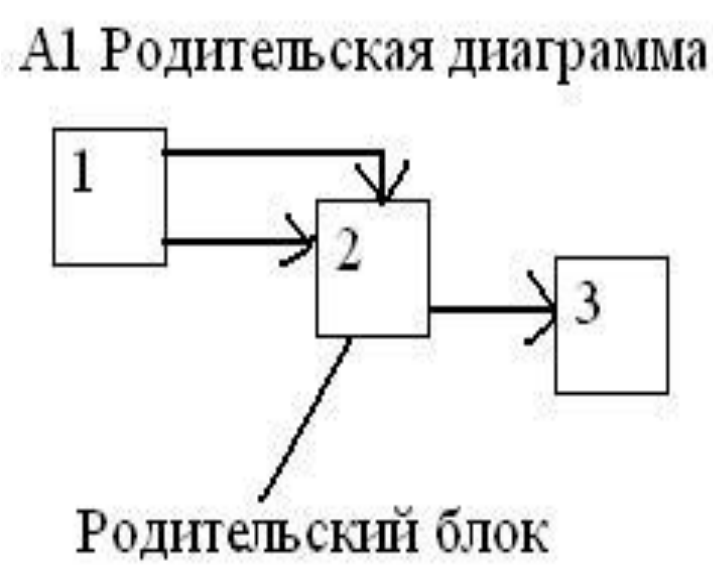
1. *Графическое представление блочного моделирования.*
2. *Строгость и точность.*

Правила SADT включают:

- ограничение количества блоков на каждом уровне декомпозиции (3-6).
- связность диаграмм (номера блоков), уникальность меток наименований (отсутствие повторяющихся имен).
- синтаксические правила для графики, разделение входов и управлений (правило определения роли данных).
- исключение влияния организационной структуры на функциональную модель.

Состав функциональной модели. Состоит из диаграмм, фрагментов текста и глоссария, имеющих ссылки друг на друга. Главный компонент диаграмма, все функции ИС и интерфейсы на них представлены как блоки и дуги.

Иерархия диаграмм. Построение SADT-модели начинается с представления системы в виде одного блока и интерфейсных дуг функций вне системы. Далее этот блок детализируется на другой диаграмме с помощью нескольких блоков, соединенных дугами.



Типы связей между функциями.

Значимость	Тип связности	Для функций	Для данных
0	Случайная	Случайная	Случайная
1	Логическая	Функции одного и того же множества или типа	Данные одного и того же множества или типа
2	Временная	Функции одного и того же периода времени	Данные, используемые в каком либо временном интервале
3	Процедурная	Функции, работающие в одной и той же фазе или итерации	Данные, используемые во время одной и той же фазы или итерации
4	Коммуникационная	Функции, использующие одни и те же данные	Данные, на которые воздействует одна и та же деятельность
5	Последовательная	Функции, выполняющие последовательные преобразования одних и тех же данных	Данные, преобразуемые последовательными функциями
6	Функциональная	Функции, объединяемые для выполнения одной функции	Данные, связанные с одной функцией

Вхождение дуг в тоннель. Применяется, чтобы показать дуги, которые более детально или полно содержание текущей диаграммы, но не является существенными для диаграмм других уровней.

Дуга «входит в тоннель» при следующих условиях:

- когда является внешней дугой, отсутствующей на родительской диаграмме (имеет скрытый источник).
- касается стороны блока, но не появляется на диаграмме, которая его декомпозирует (имеет скрытый приемник).

6.2.4 Методология описания и моделирования процессов.

Метод описания процессов IDEF3. Это методология описания процессов, рассматривающая последовательность их выполнения и причинно-следственные связи между ситуациями и событиями для структурного представления знаний о системе, а так же описания изменения состояний объектов, являющихся составной частью описываемых процессов. IDEF3 предоставляет инструментарий для наглядного исследования и моделирования сценариев выполнения процессов.

Фиксирует следующую информацию о процессе:

- объекты, которые участвуют при выполнении процесса.
- роли, которые выполняют эти объекты.
- отношения между работами в ходе выполнения сценария процесса.
- состояния и изменения, которым подвергаются объекты.
- время выполнения и контрольные точки синхронизации работ.
- ресурсы, необходимые для выполнения работ.

Описание IDEF3. В стандарте IDEF3 существует два типа диаграмм, представляющих описание одного и того же сценария технологического процесса в разных ракурсах.

- Диаграммы описания последовательности выполнения процессов (Process Flow Description Diagrams, PFDD).
- Сеть изменений объектных состояний (Object State Transition Network, OSTN).

Основные элементы диаграмм описания последовательности процессов. Графические элементы, которые включают диаграммы процесса, включают модуль полей поведения (UOB), связей старшинства, перекрестков, объектов ссылки и примечаний.

Функциональный элемент (UOB). Состоит из *имени* процесса, который отражает всевозможные ситуации которые могут произойти в моделируемой системе (процессы, функции, действия, акты, события, сценарии, процедуры, операции, решения); *номера идентификатора*, назначаемого последовательно; *ссылки* – либо на элемент модели IDEF0, либо на конкретных исполнителей.

Элемент связи. Используются для обозначения отношений между элементами UOB. Для отображения временной последовательности используются *связи старшинства* и *относительные связи*. Для описания специфических отношений используются *сдерживаемые связи старшинства.* Номера стрелок содержат префикс «L» и номер.

Связи старшинства. Выражают временные отношения старшинства между элементами диаграммы. При этом первый элемент должен завершиться, прежде чем начнет выполняться второй.

Сдерживаемые связи старшинства. В дополнение к семантике запуска связей простого старшинства указывают:

- элементу А должен предшествовать элемент В.
- элементу В должен предшествовать элемент А.

Относительные связи. Указывают, что между элементами диаграммы описания процесса существуют отношения неопределенного типа.

Связь «поток объектов». Означает, что между UOB-элементы происходит передача объекта (ов), причем первый элемент должен завершиться прежде, чем начнется второй.

Перекресток. Отображают логику отношений между множеством событий и временную синхронизацию активизации элементов. Различают перекрестки для слияния (Fan-in Junction) и разветвления (Fan-out Junction) стрелок. В PFDD-диаграмме перекрестки нумеруются и имеют префикс «J».

Типы перекрестков.

- & - логический И.
- О – логический ИЛИ.
- Х – логический перекресток НЕЭКВИВАЛЕНТНОСТИ.

Разделение на синхронные и асинхронные позволяют учитывать в диаграммах описания процессов синхронизацию времени активизации.

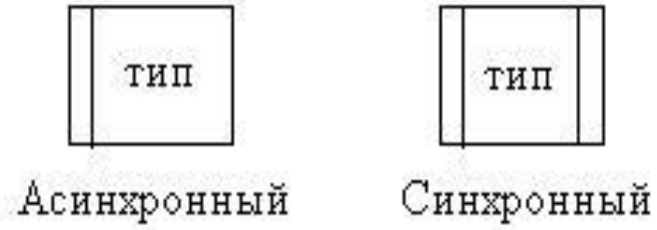
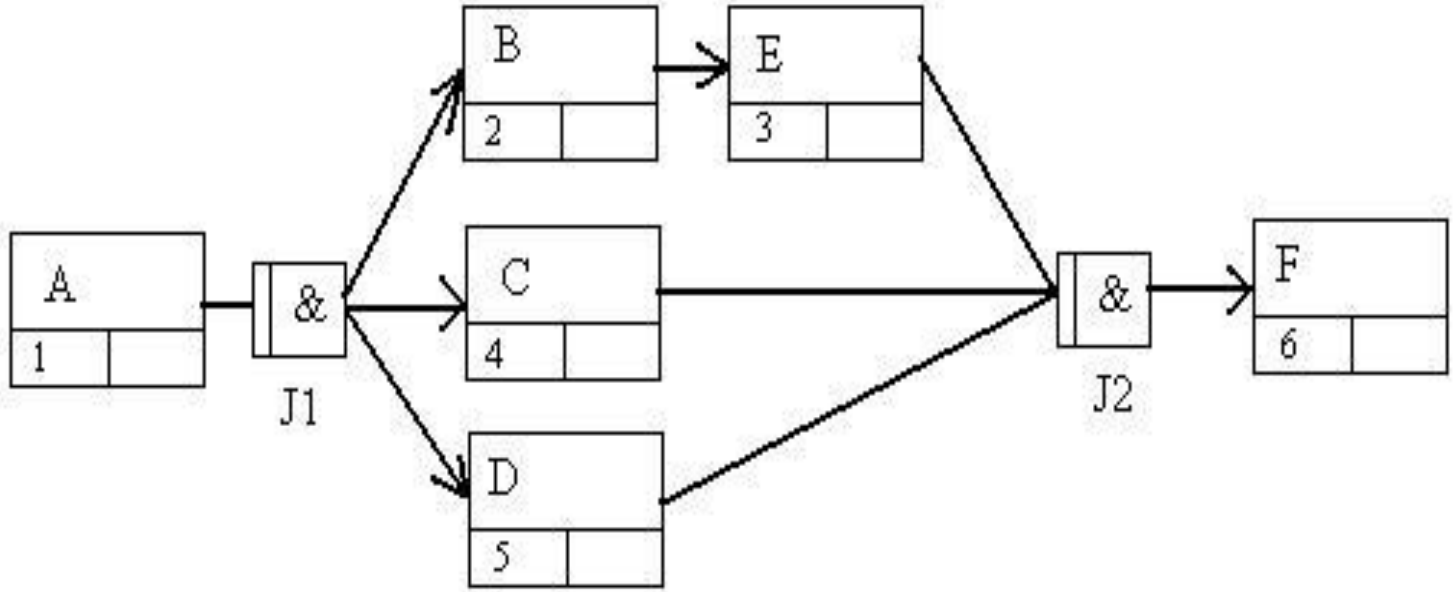
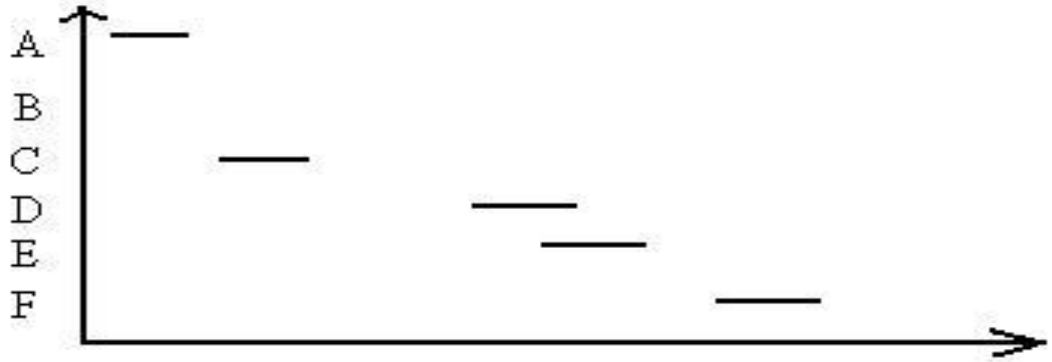


График запуска – визуальное отображение временной последовательности выполнения UOB-элементов.

Использование перекрестка «Асинхронный AND»:



Возможный график запуска:



Элемент «референт». Это элемент ссылки, расширяющие границы понимания диаграммы и упрощают конструкцию описания.

Предназначены для:

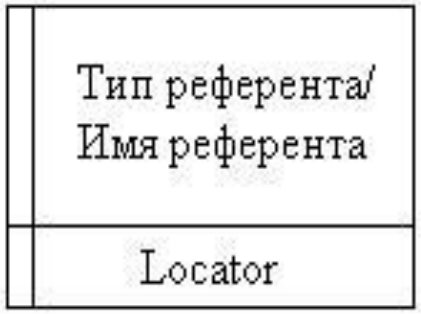
- обращения к предварительно определенному UOB-элементу без дублирования его определения.
- передачи управления или организации возвратных циклов.
- организации связи между OSTN и PFDD объектами диаграмм.

Использование референтов в диаграмме:

Тип референта	Обозначение референта	Locator
UOB	Имя UOB-элемента	Номер UOB
SCENARIO	Название сценария	Номер Scenario
TS (Transition Schematic)	Название диаграммы перехода состояний	Номер диаграммы перехода
GO-TO	Имя UOB-элемента	Номер сценария или декомпозиции, в котором находится номер UOB

Виды референтов. Определено два вида по способу запуска:

- «запустить и продолжить» - указывает, что он референт должен лишь запуститься раньше, чем выполнение элемента IDEF3, вызывающего элемент «референт», будет завершено.
- «запустить и ждать» - референт должен запуститься и



Использование референта «запустить и продолжить». Если используется референт, который имеет тип UOB, SCENARIO, GO-TO, то на выходе такого элемента не может использоваться стрелка старшинства.

UOB-референт. Если референт UOB используется в диаграмме PFDD, то во время выполнения вызывающего UOB-элемента осуществляется запуск соответствующего UOB- элемента. Если UOB-референт приложен к стрелке связывающей элементы OSTN диаграммы, то выполнение UOB-элемента должно начаться прежде, чем начнет изменяться состояние объекта.

SCENARIO-референт. Если референт используется в диаграмме PFDD, то во время выполнения вызывающего UOB-элемента осуществляется запуск вызванного сценария. Если референт приложен к стрелке связывающей элементы OSTN диаграммы, то выполнение UOB-элемента должно начаться прежде, чем начнет изменяться состояние объекта.

Использование референта «запустить и ждать». Элемент может являться источником для связи старшинства. Особенностью является невозможность использования GO-TO-референта.

Элемент «примечание». Может быть приложен к UOB-элементу, перекрестку, связи , референту. Предназначен для:

- идентификации и подчеркивания участия специфических объектов или отношений, связанных с элементом UOB, связью или переходом.
- присоединения примеров, объектов и т.д.
- отображения специальных условий, уточнения соединения или ограничения, связанных с элементами диаграмм.

Иде состоит их номера элемента, для к ечание, и номера примечания с п

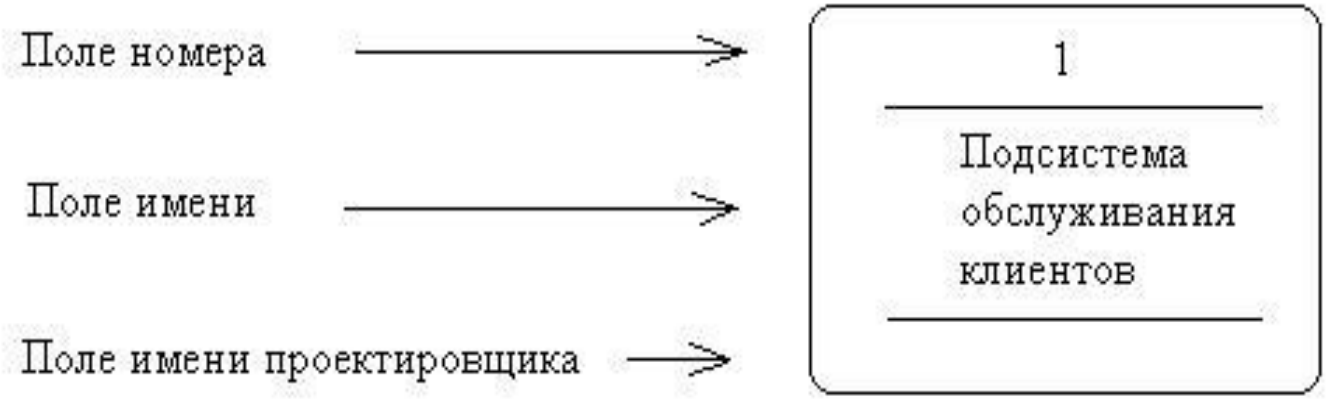
Декомпозиция процесса. Процесс создания иерархически организованной совокупности диаграмм, детализирующих определенный элемент. Результатом является описание, которое представляет собой дробление родительского элемента на меньшие и более частные операции или функции – элементы потомки. Так же включает и синтез.

6.2.5 Моделирование потоков данных (процессов).

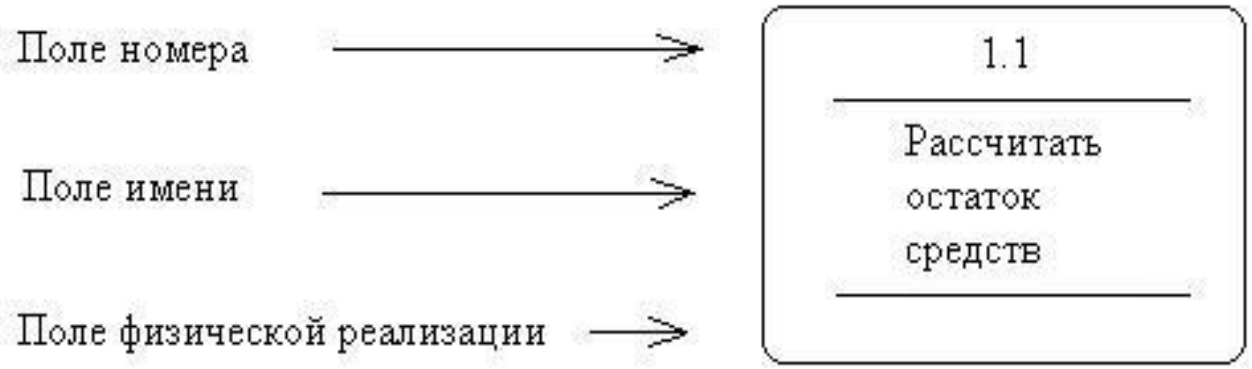
В основе данной методологии (Gane/Sarson) лежит построение модели анализируемой ИС – проектируемой или реально существующей. Определяется как иерархия диаграмм потоков данных (DFD), описывающая асинхронный процесс преобразования информации от ее ввода в систему до выдачи пользователю. Основные процессы определяют диаграммы верхних уровней иерархии (контекстные), детализирующиеся при помощи диаграмм нижнего уровня. Основными компонентами ДПД являются: внешние сущности, системы (подсистемы), процессы, накопители данных, потоки данных.

Внешние сущности. Материальный предмет или физическое лицо, представляющее собой источник или приемник информации. Находится за пределами анализируемой ИС, но при необходимости может вноситься в ее границы.

Системы и подсистемы.



Процессы. Представляет собой преобразование входных потоков данных в выходные в соответствии с определенным алгоритмом.



Накопители данных. Абстрактное устройство для хранения информации. Идентифицируется буквой «D» и произвольным числом. В общем случае является прообразом будущей базы данных.

Потоки данных. Определяет информацию, передаваемую через некоторое соединение от источника к приемнику.

Иерархия диаграмм потоков данных (ДПД). Первым шагом при построении иерархии диаграмм ДПД является построение контекстных диаграмм. Иерархия контекстных диаграмм определяет взаимодействие основных функциональных подсистем как между собой, так и с внешними входными и выходными потоками данных и внешними объектами. Далее модель следует проверить на полноту исходных данных об объектах системы и на изолированность объектов.

При детализации должны выполняться следующие правила:

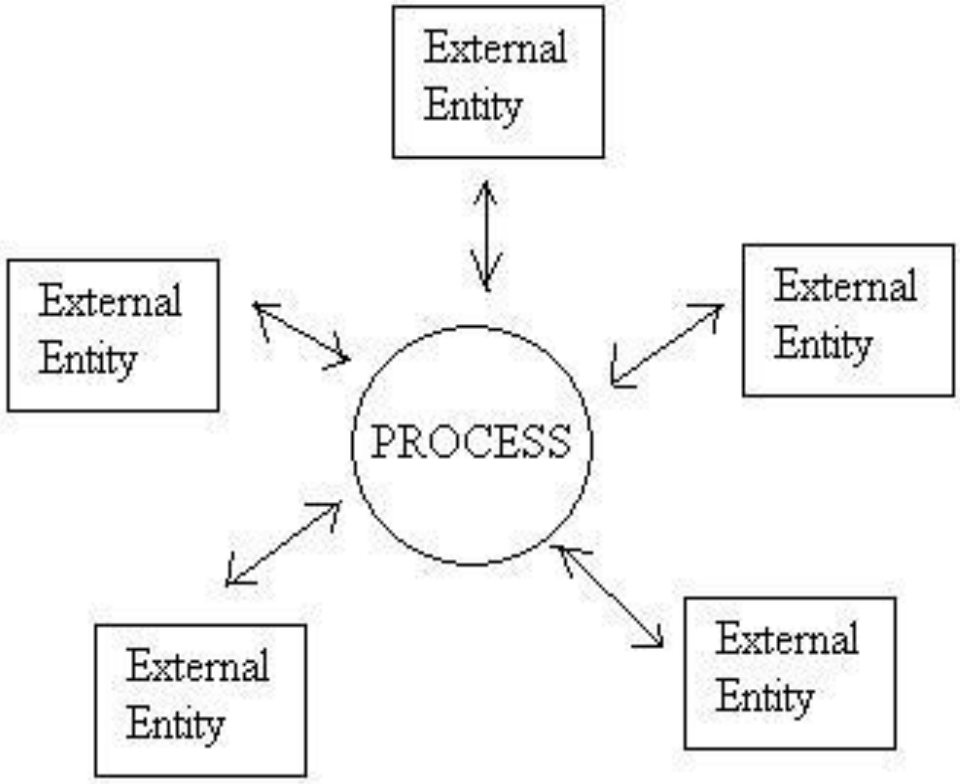
- правило балансировки – детализирующая диаграмма в качестве внешних источников/приемников данных только те компоненты, с которыми имеет информационную связь детализируемая подсистема на родительской диаграмме.
- правило нумерации – должна поддерживаться иерархическая нумерация, например, процессы, детализирующие процесс с номеров 12, получают номера 12.1, 12.2, 12.3 и т.д.

Решение о завершении детализации процесса и использовании миниспецификации (описание логики процесса) применяется аналитиком исходя из следующих критериев:

- наличие у процесса относительно небольшого количества входных/выходных потоков данных (2-3); возможности описания преобразования данных процессом в виде последовательного алгоритма.
- выполнение процессом единственной логической функции преобразования входной информации в выходную; применение спецификации небольшого объема (20-30 строк).

При построении иерархии ДПД переходить к детализации процессов следует только после определения содержания всех потоков и накопителей данных, которое описывается при помощи структур данных. Структуры данных конструируются из элементов данных и могут содержать альтернативы, условные вхождения и итерации, так же может указываться тип данных (непрерывные, дискретные).

Контекстная диаграмма верхнего уровня определяет границы модели, и как правило имеет звездообразную топологию.



6.2.6 Спецификации управления.

Спецификации управления предназначены для моделирования и документирования аспектов систем, зависящих от времени совершения события или реакции на событие. Для этой цели обычно используются диаграммы переходов состояний (STD).

Они состоят из следующих объектов:

Состояние – может рассматриваться как условие устойчивости для системы. Имя состояния должно отражать реальную ситуацию, в котором находится система, например, НАГРЕВАНИЕ, ОХЛАЖДЕНИЕ и т.д.

Начальное состояние – STD имеет только одно начальное состояние, соответствующе состоянию системы после ее инсталляции.

Переход – определяет перемещение моделируемой системы из одного состояния в другое. Событие состоит из управляющего потока (сигнала), возникающего при выполнении некоторого условия (КНОПКА НАЖАТА). Если в условии участвует входной управляющий поток управляющего процесса предка, то имя потока должно быть заключено в кавычки («ПАРОЛЬ» = 777). Кроме условия, с переходом может быть связанно действие или ряд действий, если оно необходимо для выходного управляющего потока, то его имя должно заключиться в кавычки («ВВЕДЕННАЯ КАРТА» - TRUE).

Применяются два способа построения STD. Первый способ заключается в идентификации всевозможных переходов и дальнейшем исследовании всех небессмысленных связей между ними. По второму способу сначала строится начальное состояние, потом следующие за ним и т.д.

В ситуации, когда число состояний и (или) переходов велико, для проектирования спецификаций управления используются таблицы и матрицы перехода.

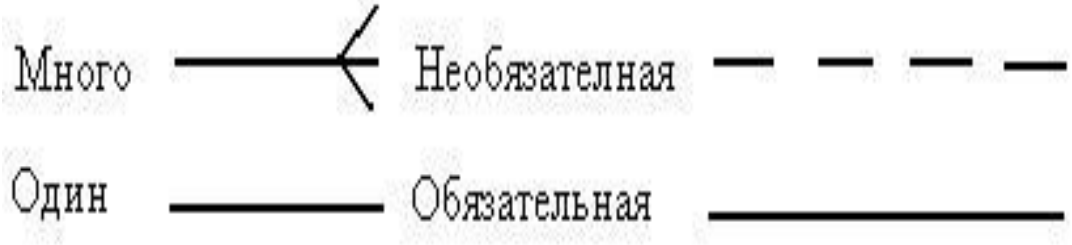
Текущее состояние	Условие	Действие	Следующее состояние
Начальное состояние	Активизируется каждый раз		ОЖИДАНИЕ
ОЖИДАНИЕ	Введенная кредитная карта	Получить пароль	ОБРАБОТКА
ОБРАБОТКА	Некорректный пароль	Удалить кредитную карту	ОЖИДАНИЕ
ОБРАБОТКА	Корректный пароль	Обеспечить требуемый сервис, удалить карту	ОЖИДАНИЕ

6.2.7 Моделирование данных.

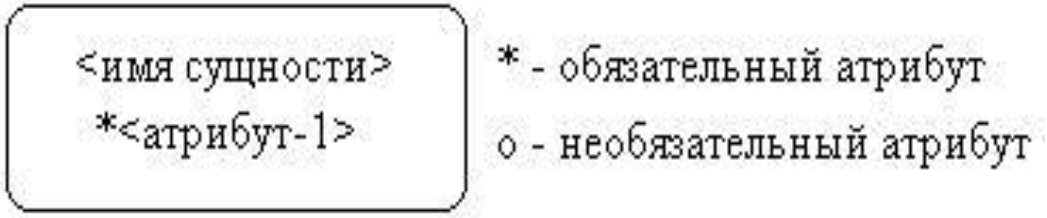
CASE-метод Баркера. Наиболее распространенным средством моделирования данных являются диаграммы «сущность-связь» (ERD).

Сущность (Entity) – Реальный или воображаемый объект, имеющий существенное значение для рассматриваемой предметной области, информация о котором подлежит хранению.

Связь (Relationship) – поименованная ассоциация между двумя сущностями, значимая для рассматриваемой предметной



Атрибут – любая характеристика сущности, значимая для рассматриваемой предметной области и предназначенная для квалификации, идентификации, классификации, количественной характеристики или выражения состояния сущности. Может быть обязательным, значит, что атрибут не может принимать неопределенных значений, или необязательным. Может быть описательным (обычным дескриптором сущности) и может входить в состав уникального



Подтипы и супертипы, когда одна сущность является обобщающим понятием для группы подобных сущностей.

Взаимно исключающие связи, определяют участие каждого экземпляра сущности только в одной связи из группы взаимно исключающих связей.

Рекурсивная связь предполагает связанность сущности с самой собой.

Неперемещаемая связь, когда экземпляр сущности не может быть перенесен из одного экземпляра связи в другой.

Метод моделирования данных IDEF1. Позволяет построить модель данных, эквивалентную реляционной модели в третьей нормальной форме. Существует усовершенствованная методология – IDEF1X, разработанная с учетом таких требований, как простота изучения и возможность автоматизации.

Сущность является независимой, если каждый экземпляр сущности может быть однозначно идентифицирован без определения его отношений с другими сущностями. Называется зависимой, если однозначная идентификация экземпляра ой сущности.

Связь может дополнительно определяться с помощью указания степени или мощности (количества экземпляров сущности-потомка, которое может существовать для каждого экземпляра сущности-родителя).

Идентифицирующая связь между сущностью-родителем и сущностью-потомком изображается сплошной линией. Пунктирная линия изображает *неидентифицирующую связь.*

Атрибуты изображаются в идее списка имен внутри блока сущности. Атрибуты, определяющие первичный ключ, размещаются наверху списка и отделяются от других горизонтальной чертой. Сущности могут так же иметь *внешние ключи* (Foreign Key, FK).

6.8.2 Сравнительный анализ структурных методологий.

Поскольку в настоящее время практически нет альтернативы ERD и STD для соответственного информационного и поведенческого моделирования, интерес представляет сравнительный анализ средств функционального моделирования, а именно DFD- и SADT-диаграмм.

Адекватность. Методология SADT успешно работает только для реорганизации хорошо специфицированных стандартизованных западных бизнес-процессов. В российской действительности с ее слабой типизацией бизнес-процессов, разумнее ориентироваться на методологию организации потоков информации и отношений.

В информационно-технологическом консалтинге, где методологии применяются к системам обработки информации, а не системам вообще, анализ является более прозрачным и адекватным при использовании DFD. Наличие миниспецификаций DFD-процессов нижнего уровня позволяет преодолеть логическую незавершенность SADT и построить полную функциональную спецификацию.

Согласованность. Главным достоинством любых моделей является возможность их интеграции с моделями других типов. В данном случае речь идет о согласованности функциональных моделей со средствами информационного и событийного моделирования. Согласование SADT-модели с ERD и (или) STD практически невозможно или носит тривиальный характер. В свою очередь DFD, ERD и STD взаимно дополняют друг друга.

Интеграция с последующими этапами. DFD могут быть легко преобразованы в модели проектирования (структурные карты). А формальные методы преобразования SADT-диаграмм в проектные решения системы обработки информации неизвестны.

6.3 Объектно-ориентированные методологии.

В настоящее время объектно-ориентированный подход является одним из быстро развивающихся направлений в проектировании систем. Все варианты подхода объединяются несколькими основополагающими принципами:

- *инкапсуляция* – свойство, при котором объекты содержат описания атрибутов и действий одновременно.
- *наследование* – метод определения объектов, при котором объекты-потомки наследуют свойства объектов-родителей.
- *полиморфизм* – свойство объектов, при котором действия с одинаковыми именами вызывают различное поведение для различных объектов.

6.3.1 Объектно-ориентированный анализ (ООА).

Методология предложена Йорденом для проектирования больших систем. ООА состоит из пяти главных шагов и соответствующих им компонентов: схемы предметной области, схемы объектов, схемы структуры, схемы атрибутов, схемы методов.

Схема предметной области. Содержит описание ее различных частей и взаимодействий между ними, и позволяет разделить предметную область на такие части, в которых содержатся однотипные объекты. Все части предметной области пронумерованы и связанны между собой.

Схема объектов. Каждому объекту на схемах соответствует графический элемент. В верхней части указывается - имя, в средней - атрибуты, в нижней – методы. Выбор набора базовых объектов во многом определяет структуру и качество всего проекта.

Схема структуры. Описание структуры предполагает определение отношений наследования двух типов: отношение вида «является» обозначаются простыми соединительными линиями; вида «состоит из» - линиями со стрелками.

Схема атрибутов. Графическая схема атрибутов повторяет схему структуры, но для каждого объекта указываются его атрибуты.

Схема методов. Графическая схема методов повторяет схему атрибутов, но для каждого объекта указывается его методы поведения.

6.3.2 Универсальный язык моделирования.

UML – это универсальный язык моделирования, разработанный в фирме Rational Software.

Пакеты как средство работы с большими проектами. Пакеты представляют собой универсальное средство для группирования элементов моделей. Пакеты могут вкладываться друг в друга и могут содержать пакеты или элементы

Рекомендации по составлению пакетов и классов в них:

1) Каждый пакет должен быть максимально независим от других пакетов, все связи должны быть локализованы и сведены к минимуму.
2) Структура пакетов должна отражать структуру предметной области.
3) Каждый пакет должен содержать классы, однотипные с точки зрения предметной области.
4) Желательно, иерархия наследования начиналась с одного базового объекта в каждом пакете, допустимо два, три, но не более.
5) Базовый объект в каждом пакете – это следующий принципиальный момент после самих пакетов.

Диаграммы классов и объектов. Диаграмма классов, это набор классов, типов данных, интерфейсов и отношений между ними. Диаграмма объектов – набор экземпляров классов и типов данных, наиболее типичным ее использованием является представление примеров структур данных.

Классы.

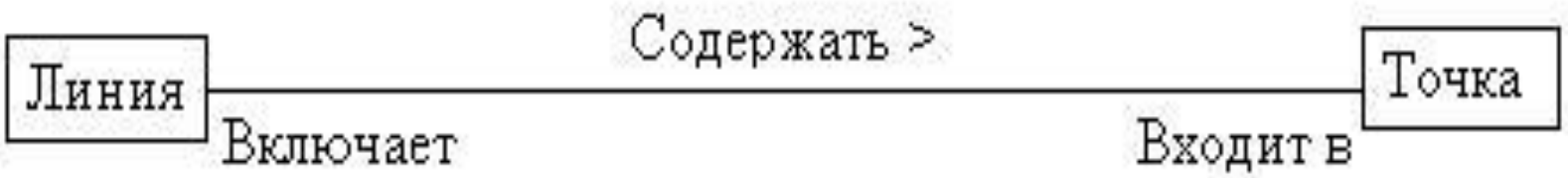
Атрибут представляется в следующем виде: *видимость имя : тип = начальное значение.*

Метод: *видимость имя (список параметров) : тип возвращаемого значения.*

Список параметров представляет собой перечень описателей параметров, разделенных запятой, имеет вид: *вид имя : тип = значение по умолчанию.*

Интерфейсы. Предназначены для спецификации внешнего вида операций для классов.

Отношения между классами. Двуместная связь – отношение между двумя классами. Изображается сплошной линией, конец которой соединенный с классом, называется ролью.



Множественность роли определяет количество экземпляров классов, участвующих в отношении. Пример: 1, 1..*, *, 0.

Сортировка роли определяет тип упорядочения элементов для множественности более чем один. Значения: упорядочены, неупорядочены.

Символ агрегации – это ромб, находящийся между линией и классом. Обозначает что класс, вблизи которого он изображен, является накопителем для класса, на другом конце связи. Ели ромб залит черным цветом, то это усиленная агрегация, называемая композиция.

Квалификатор – это один или несколько атрибутов, значения которых обеспечивают идентификацию экземпляров класса по данной связи. Множественность указанная для роли, при наличии квалификатора указывает на количество экземпляров класса, выбираемых одним значением квалификатора.

Имя роли - произвольный идентификатор, конкретизирующий роль связи для одного из классов, указывается на линии вблизи класса.

Класс, описывающий связь. Связь между классами так же может обладать свойствами присущими некоему классу. Ассоциированный класс изображается прямоугольником и связывается с отношением, для которого он задан, штриховой

N-местная связь – связь между тремя и более классами. Обозначается ромбом, соединенным линиями с классами.

Композиция. Сборка. Описывают включения классов друг в друга. Пример:

- *сборка* – элементы входят в хэш-таблицу и образуют ее. Таблица имеет собственную логику и структуру. Может не содержать ни одного элемента. В то же время элементы могут существовать изолированно от таблицы.
- *композиция* – здание образуется набором внутренних помещений. Они не могут существовать друг без друга и вне друг друга.

Обобщение – отношение между общим и уточняющим

Зависимость – обозначает любую зависимость между любыми элементами модели, изображается штриховой линией с направлением.

Диаграммы использования предназначены для отображения внешнего функционирования проектируемой системы и ее взаимодействия с внешним миром пользователей.

Включает следующие элементы:

- Внешние пользователи (actors) – воздействия которые передают или получают информацию для системы, т.е. физические объекты разной природы.
- Блоки использования (use case) – группы функций системы, которые объединяются в единое целое для внешнего пользователя.
- Связи между блоками использования и внешними пользователями.

Виды связей:

- Взаимодействие (сплошная линия).
- Расширение – дополнение функций одного блока функциями другого (линия со стрелкой помеченная словом «extends»).
- Использование (линия со стрелкой помеченная словом «uses»).

Являются весьма популярным средством благодаря простой формулировке требований ориентированной на пользователя.

Диаграмма последовательностей. Предназначена для отображения временных зависимостей, возникающих в процессе общения между объектами. Изображается как график. По вертикали – время, по горизонтали – объекты, состоящие из элементов:

- объект – прямоугольник с записанным в нем именем объекта.
- линии жизни объекта – штрихпунктирная линия, выходящая из объекта и расположенная вдоль оси времени.
- активации – тонкий вертикальный прямоугольник, обозначающий период активной жизни объекта.
- вызова метода поведения объекта (сообщение) – стрелка между активациями объектов с именем действия.
- текстовых пометок.

Диаграмма сотрудничества.

Предназначена для описания методов взаимодействия между объектами. Сотрудничество представляет набор объектов, которые вызывают методы поведения дру друга. Диаграмма сотрудничества описывает статическую структуру объектов, участвующих в реализации поведения.

Диаграмма состояний представляет конечный автомат и показывает последовательность состояний объекта, через которые он проходит во время своего существования под воздействием внешних событий.

Состояния. Состояния автомата соответствуют состояниям объектов, в которых объект удовлетворяет некоторому условию, выполняет некоторое действие или ожидает события. Каждому событию может соответствовать вложенный автомат. Изображается прямоугольником со скошенными краями, состоящим из двух частей, в первой указывается имя, во второй отображается внутреннее действие, выполняемое в ответ на определенные события.

События. Любое действие, имеющее значение с точки зрения смены состояний автомата. Виды событий:

- условие становится истинным.
- прием внешнего сигнала от одного объекта к другому.
- запрос на выполнение метода объекта.
- истечение заданного периода времени.

Простые переходы между состояниями. Простой переход из состояния 1 в состояние 2 показывает, что объект, находящийся в состоянии 1, перейдет в состояние 2 выполнит определенные действия, когда произойдет предопределенное событие и будут истинными специфицированные условия.

Составные переходы между событиями. Предназначены для отражения операций распараллеливания и синхронизации. Может иметь несколько входных и выходных состояний. После выполнения составного перехода все его выходные состояния становятся активными, а выходные наоборот.

Вложенные автоматы. Если состоянию соответствует вложенный автомат, то после выполнения действия entry для состояния начинает работать вложенный автомат. Когда достигается одно из конечных состояний вложенного автомата, то состояние считается законченным, выполняются действия по выходу (exit), и автомат готов к смене состояния под воздействием внешних воздействий. Одному состоянию может соответствовать либо одни вложенный автомат, либо несколько конкурирующих.

Диаграммы действия. Показывают поток управления, внутренний для операции, в противоположность показу реакции на внешние события (как в диаграмме состояний).

Действия. Показывает выполнение некоторой неделимой операции.

Условия. Предназначены для обозначения возможности условной передачи управления в соответствии со значением некоторого логического выражения.

Переходы. Имеют тот же смысл, что и в автомате диаграммы состояний. Но не помечаются никаким событием и имеют условие только для специальных состояний «условие».

Полосы выполнения. Диаграмма действий может быть разделена на полосы, которые включают определенный набор действий и переходов. Позволяет группировать действия в единое целое.

Отношения между действиями и объектами. Все действия выполняются над объектами. Целесообразно показать на диаграмме действий отношения между объектами и операциями. Различают два вида отношений:

- объект отвечает на выполнение операции.
- атрибуты объекта используются для выполнения операции.

Диаграммы компонентов. Отражает зависимости составных частей программного обеспечения, состоит из компонентов и отношений (зависимость, композиция).

Диаграммы развертывания. Показывает конфигурацию исполняемой программной системы, включающей в себя программные компоненты, процессы объекты. Состоит из узлов и отношений взаимодействия между узлами и компонентами. Узлы могут включать компоненты и объекты.

6.4 Практическое применение методологий проектирования ИС.

Отношение бизнес-процессов формируется при помощи нотаций и инструментальной среды, позволяющей отразить все указанные выше аспекты. Только в этом случае модель окажется полезной для предприятия, так как ее можно будет подвергнуть анализу и реорганизации.

6.4.1 Структурное моделирование ИС средствами BPwin и ERwin.

BPwin позволяет аналитику создавать сложные модели БП при минимальных усилиях. Поддерживает тир методологии – IDEF0, IDEF3, DFD.

ERwin - средство концептуального моделирования БД, использующее методологию IDEF1X.

Основные этапы построения IDEF0-модели:

1) Определить функциональную структуру.
2) Выделить для каждой функции входы, выходы, управление и механизмы.
3) Построить контекстную диаграмму.
4) Создать диаграммы декомпозиции.
5) Получить диаграмму дерева узлов.
6) Выделить «центры затрат».
7) Назначить величины затрат работам на нижнем уровне детализации по каждому центру затрат и временные характеристики работ.
8) Сформировать отчет по проведенному стоимостному анализу, указав соответствующие параметры генерации.

Рассмотрим укрупненную модель управления компанией (A0):

№	**Название работы (Activity Name)**	**Определение работы (Activity Definition)**
1	Стратегическое управление	Анализ среды организации, определение миссии, стратегии фирмы, поведения ан рынке
2	Организационное управление	Определение структуры организации
3	Экономическое управление	Экономический анализ, учет, управление финансами, издержками, себестоимостью продукции
4	Логистическое управление	Управление закупками, выпуском продукции, транспортировкой, распределением складом
5	Управление маркетингом	Исследования рынка, продукции, ценообразования, распределения
6	Управление персоналом	Управление человеческими ресурсами: прием сотрудников, регистрация, учет, обучение, оплата труда, организация труда.

Декомпозируем работу «Организационное управление»:

№	**Название работы (Activity Name)**	**Определение работы (Activity Definition)**
1	Организационная структура	Проведение описания, анализа, регламентации, совершенствования оргструктуры
2	Финансовая структуризация	Разработка и определение системы объединения финансового результата и распределения затрат
3	Управление системой качества	Разработка, внедрение, сопровождение системы качества
4	Управление администрированием	Разработка системы делопроизводства, исполнительской дисциплины, организация труда руководителей.
5	Управление правовым обеспечением управления	Проведение правовой экспертизы документов, обеспечение правовой консультации и разработка локальных правовых актов

BPwin поддерживает ABC-модель оценки затрат функционирования информационной системы:

Центр затрат	Определение
Управление	Затраты на управление, связанные с совершенствованием оргструктуры, доведением решений до подразделений, обучением персонала, внедрением и сопровождением системы качества, делопроизводством, правовой экспертизой
Человеческий ресурс	Затраты на оплату труда аналитиков, менеджеров, проводящих анализ
Обеспечение	Затраты на обеспечение возможности проведения анализа и разработок, на обеспечение необходимой информацией

6.4.1.2 BPwin – средство моделирования процессов (IDEF3).

Основные этапы построения IDEF3-модели:

1) Определить очередность запуска процессов.
2) Выделить перекрестки для отражения слияния/разветвления действий.
3) Построит возможные графики запусков процессов.
4) Построить контекстную диаграмму и диаграммы декомпозиции.
5) Создать сценарии.

Декомпозируем процесс «Описание организационной структуры»:

№	Название процесса (Activity Name)	Определение процесса (Activity Definition)
1	Сбор информации об оргструктуре	Анкетирование, интервьюирование, изучение документов
2	«Вертикальное» описание оргструктуры	Разработка классификаторов, матриц проектов
3	Актуализация описания оргструктуры в соответствии с изменениями в организации	Необходимо проводить коррекцию информации из-за постоянных изменений в организации
4	Представление проектов оргструктуры на утверждение руководству	Согласование проектов с руководством
5	Доведение «Положения о подразделениях» до структурных подразделений	Применение решений на низших уровнях управления

6.4.1.3 BPwin – средство моделирования потоков данных (DFD).

Основные этапы построения DFD-модели:

1) Выделить процессы, внешние сущности и связывающие их потоки данных.
2) Построить контекстную диаграмму.
3) Выделить процессы, определить потоки данных и хранилища на соответствующем уровне декомпозиции.
4) Построить диаграммы декомпозиции.
5) Предложит управляющие процессы, потоки и хранилища.

Декомпозируем процесс «Осуществить организационное управление»:

№	**Название работы (Activity Name)**	**Определение работы (Activity Definition)**
1	Осуществить организационную структуризацию	Провести описание, анализ. Регламентацию, совершенствовать оргструктуру
2	Осуществить правовое обеспечение управления	Разработать и определить систему объединения финансового результата и распределения затрат.
3	Обеспечить управление системой качеством	Разработать, внедрить, обеспечить сопровождение системы качества
4	Осуществить управление администрированием	Разработать систему делопроизводства, исполнительской дисциплины, организация труда руководителей.
5	Осуществить финансовую структуризацию	Провести правовую экспертизу документов, организовать обеспечение правовой консультации и разработать локальные правовые акты

Наименования хранилищ:

Название хранилища (Data Store)	**Вид хранилища**
Организационная структура	Автоматизированное
Локальные нормативные акты	Автоматизированное
БД исполнительской дисциплины	Автоматизированное
Финансовая структура	Автоматизированное
Система качества	Автоматизированное
Существующая финансовая система	Не автоматизированное
Существующая система качества	Не автоматизированное

6.4.1.4 – средство информационного моделирования (IDEF1X).

Основные этапы построения IDEF1X-модели:

1) Построение логической модели данных:

- определение сущностей, определение зависимостей между ними.
- задание первичных и альтернативных ключей.
- определение атрибутов сущностей.
- приведение модели к требуемому уровню нормальной формы.
- переход к физическому описанию модели.
- задание триггеров, процедур и ограничений.
- генерация базы данных.

2) Построение физической модели данных:

- выбор целевой СУБД, выделение представления, правила валидации для колонок, значения по умолчанию, определение индексируемых полей.
- построение на базе логической модели физической модели.
- генерирование БД на языке целевой СУБД.
- создание физической модели по схеме данных уже существующей БД на языке любой поддерживаемой ERwin СУБД.

Рассмотрим одну из функций «Организационного управления» - организационную структуризацию, которая, декомпозируется на функции:

- описание оргструктуры.
- анализ оргструктуры.
- совершенствование оргструктуры.
- регламентация.

Для построения модели определим сущности, их атрибуты и связи между ними.

Определения сущностей:

Сущность	**Определение**
Функциональная структура	Информация о функциях, классификаторах, матрицах проекций и документах
Организационная структура	Сведения о должностях, подразделениях, подчинении должностей
Регламентирующие документы	Информация о дате создания, отмены, введения, изменения
Матрицы проекций	Справочник имеющихся матриц проекций
Классификаторы	Справочник классификаторов
Решения по изменению оргструктуры	Наименование решения, где применяются решения, отражает, что было до решения и что предполагается после проведения
Изменения в организации	Наименование изменения, где прошло изменение
Применение решения	Проверяется состояние выполнения решения и состояние обучения персонала при применении решения

Полученная логическая модель данных должна быть сгенерирована в физическую модель БД.

6.4.2 Объектное моделирование ИС средством Ration Rose.

Rational Rose – мощный инструмент для анализа и проектирования объектно-ориентированных программных систем, позволяющий моделировать системы до написания кода.

Диаграмма вариантов использования. Описывает функциональное назначение системы.

Цели:

- определить общие границы и контекст моделируемой предметной области
- сформулировать общие требования к функциональному поведению проектируемой системы
- разработать исходную концептуальную модель системы для ее последующей детализации
- подготовить исходную документацию для взаимодействия разработчиков системы с ее заказчиками и пользователями

Диаграмма классов. Отражает различные взаимосвязи между отдельными сущностями предметной области, такими, как объекты и подсистемы, а так же описывать их внутреннюю структуру и типы отношений.

Диаграмма состояний. Описывает все возможные состояния одного экземпляра определенного класса и возможные последовательности его переходов из одного состояния в другое, то есть моделирует все изменения состояний объекта как его реакцию на внешние воздействия.

Диаграмма последовательности. Используется для моделирования взаимодействия объектов во времени.

Диаграмма кооперации. Главная особенность заключается в возможности графически представить не только последовательность взаимодействия , но и все структурные отношения между объектами в этом взаимодействии.

Диаграмма компонентов. Описывает особенности физического представления системы. Позволяет определить архитектуру разрабатываемой системы, установив зависимости между программными компонентами, в роли которых может выступать исходный и исполняемый код.

Диаграмма развертывания. Применяется для решения задач рационального размещения компонентов, в целях эффективного использования распределенных вычислительных и коммуникационных ресурсов сети, обеспечения безопасности и других.