

**Российская академия народного хозяйства и государственной
службы при Президенте РФ**

РЕФЕРАТ ПО ИНФОРМАТИКЕ

Тема работы «Модели информационной системы»

Выполнил:

Полянский А.С.

Проверил:

Яворский В.М.

Липецк 2021

Введение.

1.Теоретическая часть.

1.1. Этапы разработки информационной системы.

Согласно схеме процесса разработки информационной системы, она включает следующие этапы:

- Выявление информационных потребностей конечных пользователей
- Концептуальное проектирование
- Разработка архитектуры ИС
- Логическое проектирование
- Отладка и тестирование прикладных программ
- Сопровождение

На первом этапе на основе анализа предметной области строится функциональный граф, связывающий функции будущей системы с входными и выходными данными. Выходные данные одной функции могут служить в качестве входных для других.

Большинство универсальных компьютеров имеют архитектуру фон-Неймана, предполагающую разделение процессов и данных. Это вынуждает разработчиков систем уже после первого этапа проектирования отделять данные от функций. Далее работы по проектированию схем данных и процессов (задач) выполняются как бы параллельно, что является источником многих противоречий.

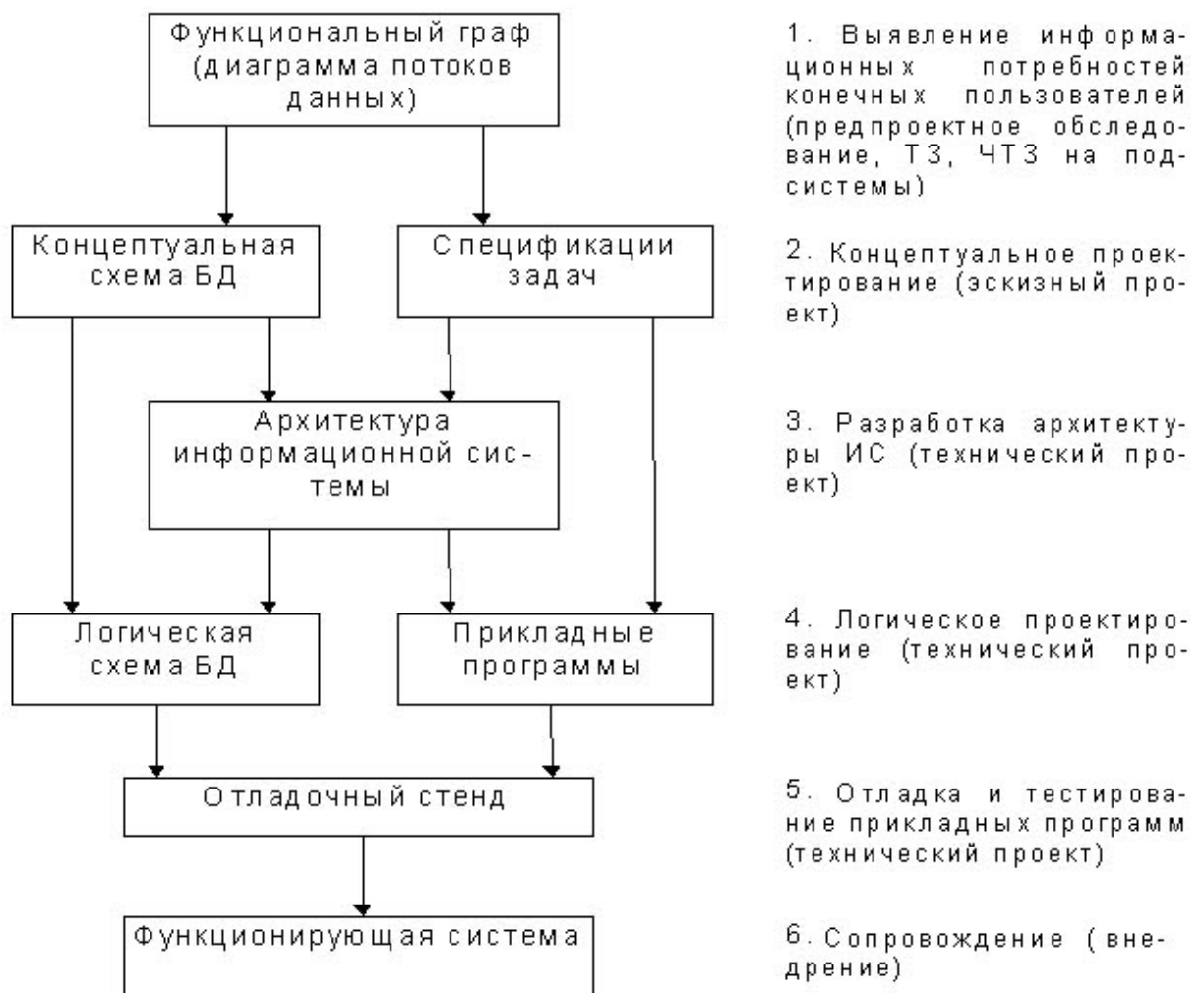
На втором этапе данные структурируются в концептуальную схему (КС) базы данных (БД), а функции объединяются в задачи будущей системы. При разработке концептуальной схемы БД проектировщик руководствуется следующими абстракциями: агрегацией, обобщением, ассоциацией. КС базы данных изображается с помощью ERD-диаграмм (диаграмм Чена или

Баркера). На этом этапе также разрабатываются спецификации будущей системы, т. е. определяются входные и выходные данные и алгоритмы связей между ними. Следует отметить, что концептуальный проект не зависит от реализации и отражает содержательную сторону будущей системы.

На этапе разработки архитектуры ИС решаются следующие задачи:

- выбирается модель доступа к данным
- осуществляется выбор структуры комплекса технических средств (КТС)
- определяется состав общесистемных пакетов (ОС, СУБД и др.)
- выполняется распределение задач по машинам распределённой информационной системы

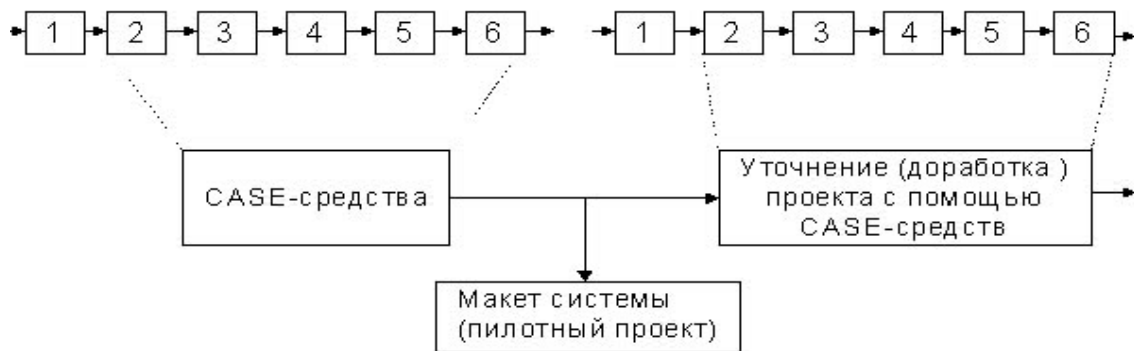
ЭТАПЫ



На этапе логического проектирования выполняется отображение концептуальной схемы базы данных и спецификаций прикладных задач в СУБД - ориентированную среду. При этом КС базы данных преобразуется в логическую схему БД, а спецификации задач - в прикладные программы на конкретном языке.

То есть CASE-средства позволяют реализовать все основные этапы жизненного цикла системы , но за более короткий промежуток времени. Несмотря на многие рекламные заявления, эти средства всё таки предназначены для профессиональных разработчиков, так как для их эффективного использования требуется знать, по крайней мере, элементы теории проектирования баз данных и иметь навыки в программировании (хотя бы на языках класса 4GL). Платой за это является упрощённый неэффективный вариант системы, получающийся в результате использования CASE-продуктов. Но важнейшее преимущество использования CASE-средств, перевешивающее указанный недостаток, состоит в том, что профессиональные разработчики могут быстро доработать проект при изменении требований к системе со стороны конечных пользователей. Основное преимущество спиральной модели заключается в том, что при её использовании время реализации витка жизненного цикла ИС намного меньше, чем при применении каскадной схемы.

При использовании схемы проектирования, представленной на рис. 4,



- 1 - выбор архитектуры (CASE-средства не поддерживают этот этап),
 2 - выявление информационных потребностей,
 3 - концептуальное проектирование,
 4 - логическое проектирование,
 5 - отладка, тестирование, анализ,
 6 - использование и сопровождение.

следует соблюдать следующие правила:

- Важно с самого начала правильно выбрать общесистемное программное обеспечение (ОС, СУБД, CASE-продукты и др.). Использование на предприятии единой платформы общесистемных средств существенно облегчает модификацию и стыковку приложений на следующих этапах разработки. Инициативные конечные пользователи могут сразу начать макетировать свою предметную область с помощью доступных общесистемных программных средств (редакторов, электронных таблиц, настольных СУБД и др.).
- Нельзя затягивать процесс такого хаотичного выявления информационных потребностей по следующим причинам:
- Часто пользователь видит только свою предметную область и многие подразделения пытаются обособиться ("мой сервер", "моя сеть", "мне больше ничего не надо"),
- Многие сотрудники инертны, не инициативны и пытаются использовать только простые средства (текстовый редактор).
- Разработка проекта должна вестись профессиональными разработчиками в контакте с конечными пользователями.

1.2. Общие сведения о моделях информационной системы.

Для создания И.С. существует два подхода:

1)Объектно-ориентированный

2)Структурный

Объектно-ориентированный подход использует объектную декомпозицию, при этом статическая структура системы описывается в терминах объектов и связей между ними, а поведение системы описывается в терминах обмена сообщениями между объектами. Каждый объект системы обладает своим собственным поведением, моделирующим поведение объекта реального мира. Понятие "объект" впервые было использовано около 30 лет назад в технических средствах при попытках отойти от традиционной архитектуры фон Неймана и преодолеть барьер между высоким уровнем программных абстракций и низким уровнем абстрагирования на уровне компьютеров. С объектно-ориентированной архитектурой также тесно связаны объектно-ориентированные операционные системы. Однако наиболее значительный вклад в объектный подход был внесен объектными и объектно-ориентированными языками программирования: Simula, Smalltalk, C++, Object Pascal. На объектный подход оказали влияние также развивавшиеся достаточно независимо методы моделирования баз данных, в особенности подход "сущность-связь". Концептуальной основой объектно - ориентированного подхода является объектная модель. Основными ее элементами являются: • абстрагирование (abstraction); • инкапсуляция (encapsulation); • модульность (modularity); • иерархия (hierarchy). Кроме основных имеются еще три дополнительных элемента, не являющихся в отличие от основных строго обязательными: • типизация (typing)', • параллелизм (concurrency)', • устойчивость (persistence). Абстрагирование это выделение существенных характеристик некоторого объекта, которые отличают его от всех других видов объектов и, таким образом, четко

определяют его концептуальные границы относительно дальнейшего рассмотрения и анализа. Абстрагирование концентрирует внимание на внешних особенностях объекта и позволяет отделить самые существенные особенности его поведения от деталей их реализации. Выбор правильного набора абстракций для заданной предметной области представляет собой главную задачу объектно-ориентированного проектирования. Инкапсуляция — это процесс отделения друг от друга отдельных элементов объекта, определяющих его устройство и поведение. Инкапсуляция служит для того, чтобы изолировать интерфейс объекта, отражающий его внешнее поведение, от внутренней реализации объекта. Объектный подход предполагает, что собственные ресурсы, которыми могут манипулировать только методы самого класса, скрыты от внешней среды. Абстрагирование и инкапсуляция являются взаимодополняющими операциями: абстрагирование фокусирует внимание на внешних особенностях объекта, а инкапсуляция (или, иначе, ограничение доступа) не позволяет объектам-пользователям различать внутреннее устройство объекта. Объектно-ориентированный подход

Модульность — это свойство системы, связанное с возможностью ее декомпозиции на ряд внутренне связанных, но слабо связанных между собой модулей. Инкапсуляция и модульность создают барьеры между абстракциями. Иерархия — это ранжированная или упорядоченная система абстракций, расположение их по уровням. Основными видами иерархических структур применительно к сложным системам являются структура классов (иерархия по номенклатуре) и структура объектов (иерархия по составу). Примерами иерархии классов являются простое и множественное наследование (один класс использует структурную или функциональную часть соответственно одного или нескольких других классов), а иерархии объектов - агрегация. Типизация — это ограничение, накладываемое на класс объектов и препятствующее взаимозаменяемости различных классов (или

сильно сужающее ее возможность). Типизация позволяет защититься от использования объектов одного класса вместо другого или, по крайней мере, управлять таким использованием. Параллелизм — свойство объектов находиться в активном или пассивном состоянии и различать активные и пассивные объекты между собой. Устойчивость — свойство объекта существовать, во времени (вне зависимости от процесса, породившего данный объект) и/или в пространстве (при перемещении объекта из адресного пространства, в котором он был создан). Основные понятия объектно-ориентированного подхода - объект и класс. Объект определяется как осязаемая реальность (tangible entity) — предмет или явление, имеющие четко определяемое поведение. Объект обладает состоянием, поведением и индивидуальностью; структура и поведение схожих объектов определяют общий для них класс. Термины "экземпляр класса" и "объект" являются эквивалентными. Состояние объекта характеризуется перечнем всех возможных (статических) свойств данного объекта и текущими значениями (динамическими) каждого из этих свойств. Поведение характеризует воздействие объекта на другие объекты и наоборот относительно изменения состояния этих объектов и передачи сообщений. Иначе говоря, поведение объекта полностью определяется его действиями. Индивидуальность — это свойства объекта, отличающие его от всех других объектов. Определенное воздействие одного объекта на другой с целью вызвать соответствующую реакцию называется операцией. Как правило, в объектных и объектно-ориентированных языках операции, выполняемые над данным объектом, называются методами и являются составной частью определения класса. Класс — это множество объектов, связанных общностью структуры и поведения. Любой объект является экземпляром класса. Определение классов и объектов — одна из самых сложных задач объектно-ориентированного проектирования. Следующую группу важных понятий объектного подхода

составляют наследование и полиморфизм. Понятие полиморфизма может быть интерпретировано, как способность класса принадлежать более чем одному типу. Наследование означает построение новых классов на основе существующих с возможностью добавления или переопределения данных и методов. Объектно-ориентированная система изначально строится с учетом ее эволюции. Наследование и полиморфизм обеспечивают возможность определения новой функциональности классов с помощью создания производных классов — потоков базовых классов. Потомки наследуют характеристики родительских классов без изменения их первоначального описания и добавляют при необходимости собственные структуры данных и методы. Определение производных классов, при котором задаются только различия или уточнения, в огромной степени экономит время и усилия при производстве и использовании спецификаций и программного кода. Важным качеством объектного подхода является согласованность моделей деятельности организации и моделей проектируемой системы от стадии формирования требований до стадии реализации. Требование согласованности моделей выполняется благодаря возможности применения абстрагирования, модульности, полиморфизма на всех стадиях разработки. Модели ранних стадий могут быть непосредственно подвергнуты сравнению с моделями реализации. По объектным моделям может быть прослежено отображение реальных сущностей моделируемой предметной области (организации) в объекты и классы информационной системы.

Настоящее время господствующим направлением проектирования ИС является объектно-ориентированная технология как основа создания открытых, гибких, многофункциональных систем для различных предметных областей.

1.3. Подходы к разработке АИС.

На сегодняшний день существует два основных подхода к разработке информационных систем, отличающихся критериями декомпозиции. Первый подход, получивший название функционально-модульного или структурного, определяется принципом алгоритмической декомпозиции. В соответствии с этим принципом осуществляется разделение функций ИС на модули по функциональной принадлежности, и каждый модуль реализует один из этапов общего процесса. Такой традиционный функционально-модульный подход к проектированию ИС, получивший название «модель водопада», предусматривает строго последовательный порядок действий. Главный недостаток такого подхода заключается в движении информации в одном направлении (аналог по течению реки). Если при проектировании или эксплуатации возникает проблема, то она решается только на данной стадии проекта, не затрагивая предыдущих стадий. Недостаточная обратная связь приводит к ограниченным исправлениям, что в свою очередь приводит к деформированным реализациям. Ориентация на функционально-модульный подход увеличивает вероятность потери контроля над решением возникающих проблем.

Объектно-ориентированная технология проектирования ИС предоставляет мощную, гибкую, универсальную концептуальную основу для конструирования информационно-управляющих систем в различных областях хозяйственной деятельности и управления, сочетающую использование моделей современной логики, объектного подхода к компонентам предметной области, современных инструментальных средств визуального программирования и СУБД с SQL-интерфейсом.

Объектно-ориентированная технология проектирования ИС включает в себя следующие компоненты:

- технологию конструирования концептуальной объектно-ориентированной модели предметной области;

- инструментальные средства спецификации проектных решений;
- библиотеки типовых компонентов модели предметной области;
- типовые проектные решения для ряда функциональных областей.

1.4. Методология функционального моделирования SADT.

1.4.1. Состав функциональной модели.

Результатом применения метода SADT является модель, которая состоит из диаграмм, фрагментов текстов и глоссария, имеющих ссылки друг на друга. Диаграммы — главные компоненты модели, все функции организации и интерфейсы на них представлены как блоки и дуги соответственно. Место соединения дуги с блоком определяет тип интерфейса. Управляющая информация входит в блок сверху, в то время как входная информация, которая подвергается обработке, показана с левой стороны блока, а результаты (выход) показаны с правой стороны. Механизм (человек или автоматизированная система), который осуществляет операцию, представляется дугой, входящей в блок снизу (рис. 1.4.1).

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



Рис. 1.4.1. Функциональный блок и интерфейсные дуги.

На рис. 1.4.2. приведены четыре диаграммы и их взаимосвязи, показана структура SADT-модели. Каждый компонент модели может быть декомпозирован на другой диаграмме. Каждая диаграмма иллюстрирует "внутреннее строение" блока на родительской диаграмме.

1.4.2. Декомпозиция диаграмм.

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

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

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

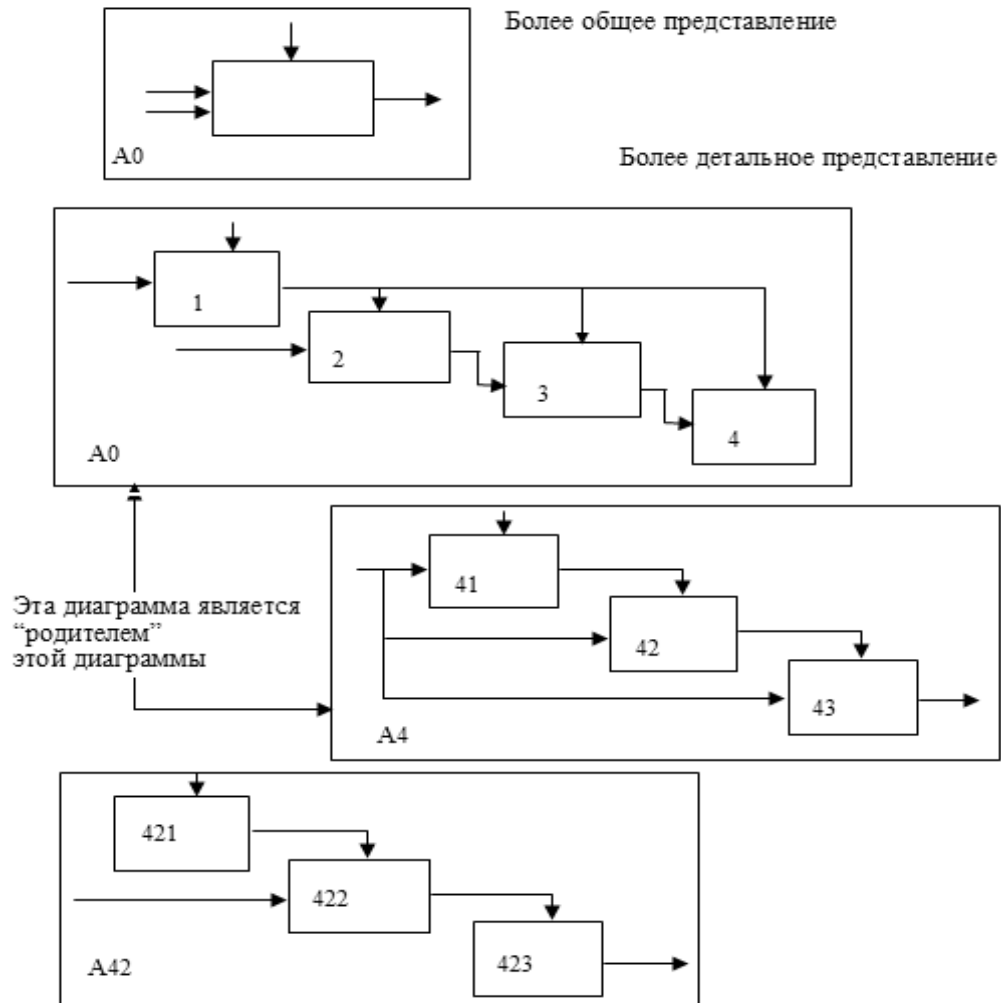


Рис. 1.4.2. Структура SADT-модели. Декомпозиция диаграмм.