

---

# Методология структурного анализа и проектирования SADT

# Методология структурного моделирования SADT

Методология SADT (Structured Analysis and Design Technique) была создана и опробована на практике в период с 1969 по 1973 гг. Автором методологии SADT является Дуглас Росс.

Предназначения для моделирования систем на основе принципов структурного анализа. Методология предлагает графический язык проектирования систем, в котором сочетаются декомпозиция и иерархическое упорядочение и для обозначения составляющих системы используется графическая конструкция, называемая SA-блок.

# Предпосылки создания SADT

- **Возрастание сложности проектируемых систем.**
- **Необходимость формализации процесса разработки при создании крупномасштабных систем.**

**Процесс разработки систем был формально разбит на этапы:**

1. **Анализ** – определение того, что система будет делать
2. **Проектирование** – определение подсистем и их взаимодействие
3. **Реализация** – разработка подсистем по отдельности
4. **Объединение** – сборка подсистем в целое
5. **Тестирование** – проверка работы системы
6. **Установка** – введение системы в действие
7. **Функционирование** – использование системы

**Данная последовательность этапов разработки стала традиционной**

# Проблемы традиционного подхода

- Неучастие пользователя в процессе разработки.
- Сложности и отсутствие согласования результатов этапов разработки.
- Сложности в качественной и количественной оценке процесса разработки.
- Трудности в выявлении ошибок, допущенных на ранних этапах разработки системы.
- Неполнота функциональных спецификаций.
- Отсутствие согласованности между спецификациями и результатами проектирования.

# Результат применения традиционного подхода

- Выявление необходимости совершенствования методов анализа как ключа к созданию систем, эффективных по стоимости, производительности и надежности.
- Поиск методологии, применение которой способно было бы преодолеть выявленные недостатки традиционного подхода.
- Появление и совершенствование методологии структурного анализа SADT.

# Преимущества SADT

- Легко отражает такие системные характеристики как управление, обратная связь и исполнители, так как возникла на базе проектирования систем общего вида в отличие от структурных методов, «выросших» из проектирования программного обеспечения.
- Имеет развитые процедуры поддержки коллективной работы.
- Применяется на ранних стадиях создания системы, что позволяет избежать наиболее дорогостоящих ошибок.
- Успешно сочетается с другими структурными методами.

**Разработка и широкое успешное использование ее графического языка превратило SADT в методологию, способную значительно повысить качество продуктов, создаваемых на ранних этапах проектов.**

# Сущность структурного подхода

Система декомпозируется (разбивается) на функциональные подсистемы до нужной степени детализации.

Базовые принципы:

- принцип «разделяй и властвуй».
- принцип иерархического упорядочивания

# Использование SADT

Методология SADT может использоваться для моделирования широкого круга систем и определения требований и функций, а затем для разработки системы, которая удовлетворяет этим требованиям и реализует эти функции. Для уже существующих систем SADT может быть использована для анализа функций, выполняемых системой, а также для указания механизмов, посредством которых они осуществляются.



Методология SADT может быть направлена как для описания функций, выполняемых системой, так и на описание объектов, составляющих систему.

В первом случае методология SADT представляет собой совокупность методов, правил и процедур, предназначенных для построения функциональной модели объекта какой-либо предметной области. Функциональная модель SADT отображает функциональную структуру объекта, т.е. производимые им действия и связи между этими действиями.

Во-втором случае методология SADT представляет собой совокупность методов, правил и процедур, предназначенных для описания объектов, входящих в систему, их свойств и взаимосвязей между ними

# Методологии SADT

- IDEF0 (Icam Definition) модели и соответствующие функциональные диаграммы.
- DFD (Data Flow Diagrams) диаграммы потоков данных.
- ERD (Entity-Relationship Diagrams) диаграммы "сущность-связь«.

# Методология функционального моделирования IDEF0

Методология функционального моделирования IDEF0 (Icam DEFinition) была разработана на основе SADT и являлась основной частью программы ICAM (Интеграция компьютерных и промышленных технологий), проводимой по инициативе ВВС США.

# Принципы функционального моделирования. Основные понятия.

- **Система** – совокупность взаимодействующих компонент и взаимосвязей между ними.
- **Моделирование** – процесс создания точного описания системы.
- **SADT модель** – полное, точное и адекватное описание системы, имеющее конкретное назначение, которое называется целью модели. SADT модель может быть сосредоточена либо на функциях системы (функциональная модель), либо на ее объектах (модель данных).
- **Цель модели** – получение ответов на некоторую совокупность вопросов относительно системы. Список вопросов сводится к одной-двум фразам, которые и формулируют цель.

- **Субъект моделирования** – сама система.
- **Границы системы** - точно определяют, что является и что не является субъектом моделирования, что входит в систему и что лежит за ее пределами. SADT-модель всегда имеет единственный субъект.
- **Точка зрения** – позиция, с которой наблюдается система и создается ее модель. Это позиция человека или объекта, в которую нужно встать, чтобы увидеть систему в действии.

В процессе моделирования **субъект** определяет, что включить в модель, а что исключить из нее. **Точка зрения** диктует выбор нужной информации о субъекте и форму ее подачи. **Цель** становится критерием окончания моделирования.

# Концепции IDEF0

- IDEF0-Модель отображает систему в виде иерархии диаграмм.
- Каждая диаграмма содержит блоки и дуги.
- Диаграмма в виде блока отображает функцию. Блоки имеют доминирование;
- Интерфейсы входа/выхода представляются дугами, входящими в блок и выходящими из него;
- Интерфейсные дуги показывают взаимодействие блоков друг с другом;
- Интерфейсные дуги выражают "ограничения", определяющие, когда и каким образом функции выполняются и управляются.

# Правила IDEF0

- Диаграмма, лежащая на вершине иерархии, называется контекстной. На этой диаграмме вся система представляется в виде единого функционального блока.
- Следующей в иерархии является диаграмма декомпозиции контекстной диаграммы. На ней функциональный блок контекстной диаграммы декомпозируется на составляющие его функциональные блоки. Каждый из этих блоков может иметь свою диаграмму декомпозиции.
- Количество блоков на каждом уровне декомпозиции ограничено (может быть от 3 до 6);
- Диаграммы связаны по номерам блоков;
- Метки и наименования уникальны;
- Входы и управления разделены по роли данных;
- Исключено влияние организационной структуры на функциональную модель.

# Состав функциональной модели IDEF0

Функциональная модель состоит из диаграмм, фрагментов текстов и глоссария, имеющих ссылки друг на друга.

Диаграммы - главные компоненты модели, все функции системы и интерфейсы на них представлены как блоки и дуги.

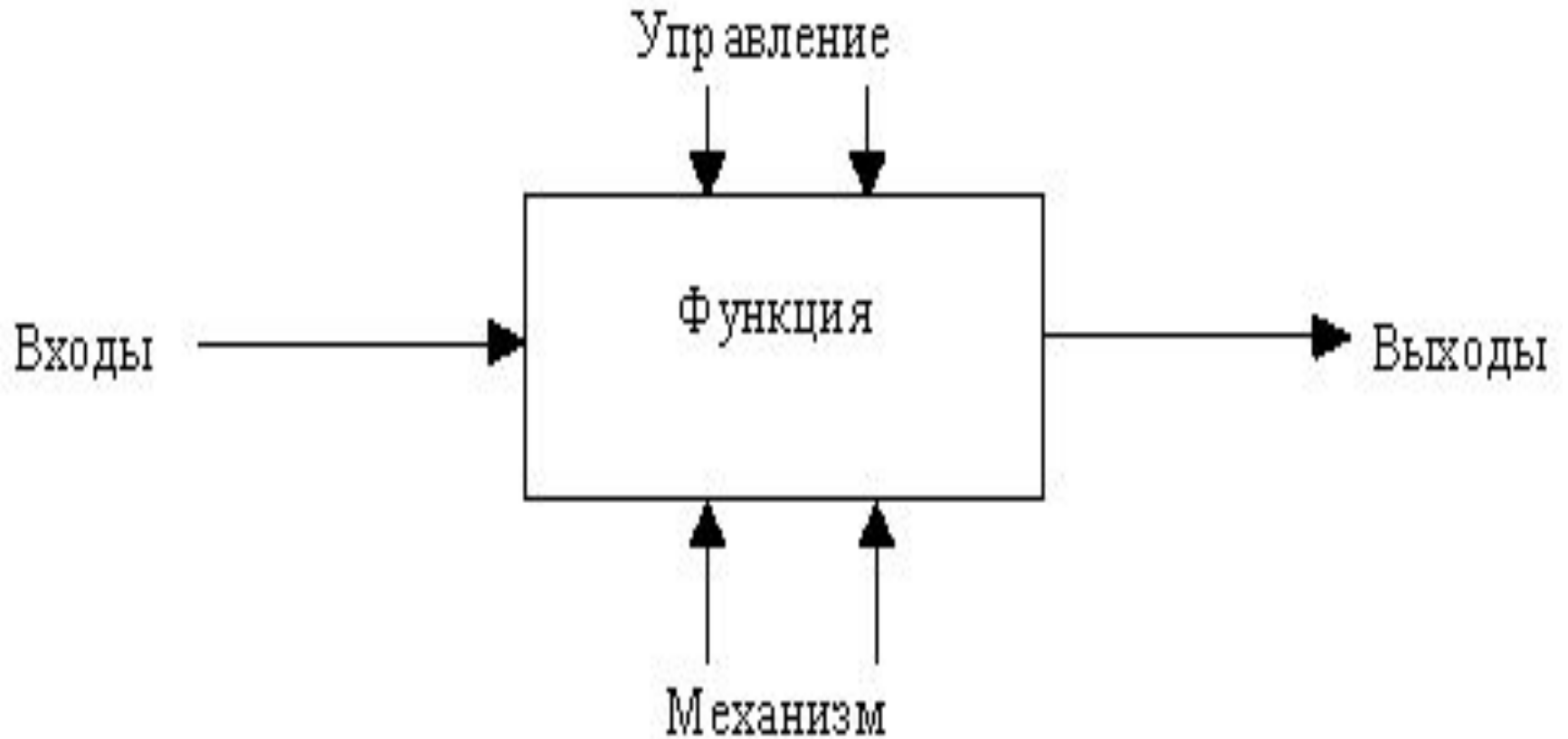
Место соединения дуги с блоком определяет тип интерфейса.

Управляющая информация входит в блок сверху, информация, которая подвергается обработке, - слева, результаты выхода - справа стороны.

Механизм (человек или автоматизированная система), который осуществляет операцию, входит в блок снизу.



# Функциональный блок и интерфейсные дуги



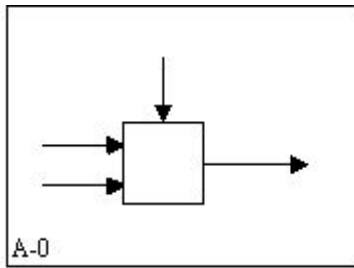
# Иерархия диаграмм

На вершине иерархии находится диаграмма, на которой система представляется в виде единого блока и дуг, изображающих интерфейсы с функциями вне системы. Контекстная диаграмма.

Уровнем ниже находится диаграмма, на которой блок, представляющий систему в целом, детализируется с помощью нескольких блоков, соединенных интерфейсными дугами. Эти блоки представляют основные подфункции исходной функции.

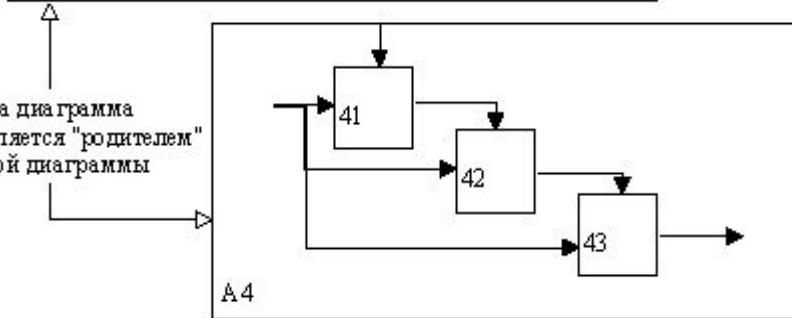
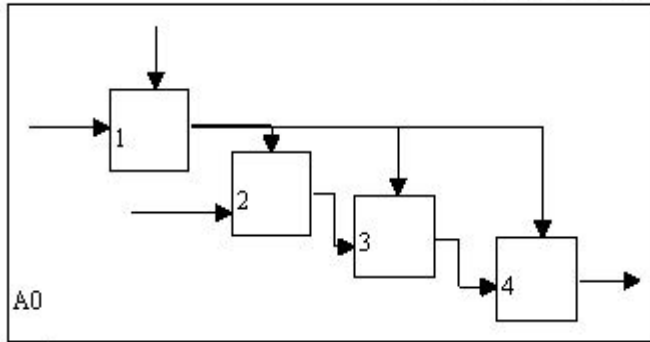
# Правила декомпозиции функциональных блоков

- Каждая функция может быть декомпозирована на подфункции;
- Подфункция может содержать только те элементы, которые входят в исходную функцию;
- Родительский блок и его интерфейсы обеспечивают контекст. Из модели нельзя выбросить какие-либо элементы или добавить их;
- Дуги, входящие в блок и выходящие из него на диаграмме верхнего уровня, являются точно теми же самыми, что и дуги, входящие в диаграмму нижнего уровня и выходящие из нее.

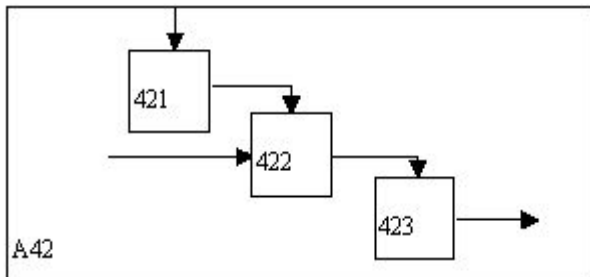


Более общее представление

Более детальное представление



Эта диаграмма является "родителем" этой диаграммы



# Структура IDEF0-модели. Декомпозиция диаграмм

- Каждый блок на диаграмме имеет свой номер.
- Для того, чтобы указать положение любой диаграммы или блока в иерархии, используются номера диаграмм. Например, A21 является диаграммой, которая детализирует блок 1 на диаграмме A2. Аналогично, A2 детализирует блок 2 на диаграмме A0, которая является самой верхней диаграммой модели.

# Иерархия диаграмм

