

# Методология SADT

- Системное проектирование - это дисциплина, определяющая подсистемы, компоненты и способы их соединения, задающая ограничения, при которых система должна функционировать, выбирающая наиболее эффективное сочетание людей, машин и программного обеспечения для реализации системы

- В современных условиях динамично развивается рынок комплексных интегрированных систем автоматизации предприятий и учреждений самого различного профиля (финансовых, промышленных, офисных) и самых различных размеров с разнообразными схемами иерархии, начиная от малых предприятий численностью в несколько десятков человек и завершая крупными корпорациями численностью в десятки тысяч сотрудников. Такие системы предназначены для решения задач как предприятия в целом (управление финансовыми ресурсами, управление запасами, планирование и производство, сбыт и снабжение, техническое обслуживание и ремонт оборудования, управление персоналом и т.п.), так и уровня его производственных подразделений, цехов и участков.

- SADT - одна из самых известных и широко используемых систем проектирования.
- SADT - аббревиатура слов Structured Analysis and Design Technique (Технология структурного анализа и проектирования) - это графические обозначения и подход к описанию систем. Дуглас Т. Росс ввел их почти 20 лет назад.

- Программное обеспечение телефонных сетей, системная поддержка и диагностика, долгосрочное и стратегическое планирование, автоматизированное производство и проектирование, конфигурация компьютерных систем, обучение персонала, встроенное программное обеспечение для оборонных систем, управление финансами и материально-техническим снабжением - вот некоторые из областей эффективного применения SADT.

- В программе интегрированной компьютеризации производства (ICAM) Министерства обороны США была признана полезность SADT, что привело к стандартизации и публикации ее части, называемой IDEFO
- Под названием IDEFO SADT применялась тысячами специалистов в военных и промышленных организациях

- В коммерческом мире SADT используется для определения требований.
- В этом качестве она конкурирует с методами, ориентированными на потоки данных, - структурного проектирования Е.Иордана, структурного анализа Т. ДеМарко, структурного системного анализа С. Гейна и Т. Сарсона, а также с методами структуризации данных - методами М.Джексона, Лж.Д. Варнира и К. Орра.

- В отличие от этих методов структурного анализа, истоки которых нужно искать в проектировании программного обеспечения, SADT создана для описания системы и ее среды до определения требований к программному обеспечению или к чему-либо другому.

- Поставив своей целью описание системы в общем, создатели SADT изобрели графический язык и набор процедур анализа для понимания системы прежде, чем можно представить себе ее воплощение.
- Таким образом, SADT, как правило, применяется на ранних этапах процесса создания системы, который часто называют "жизненным циклом системы", и иногда за этим следует применение упомянутых выше методов



# Предпосылки создания SADT

- SADT возникла в конце 60-х годов в ходе революции, вызванной структурным программированием. Когда большинство специалистов билось над созданием программного обеспечения, немногие старались разрешить более сложную задачу создания крупномасштабных систем, включающих как людей и машины, так и программное обеспечение, аналогичных системам, применяемым в телефонной связи, промышленности, управлении и контроле за вооружением. В то время специалисты, традиционно занимавшиеся созданием крупномасштабных систем, стали осознавать необходимость большей упорядоченности.

Разработчики начали формализовать процесс создания системы, разбивая его на следующие фазы:

- анализ - определение того, что система будет делать,
- проектирование - определение подсистем и их взаимодействие,
- реализация - разработка подсистем по отдельности, объединение - соединение подсистем в единое целое,
- тестирование - проверка работы системы,
- установка - введение системы в действие,
- функционирование - использование системы.

- Эта последовательность всегда выполнялась итерационно, потому что система полностью никогда не удовлетворяла требованиям пользователей, поскольку их требования часто менялись

- Традиционные подходы к созданию систем приводили к возникновению многих проблем. Не было единого подхода. Привлечение пользователя к процессу разработки не контролировалось. Проверка на согласованность проводилась нерегулярно или вообще отсутствовала. Результаты одного этапа не согласовывались с результатами других. Процесс с трудом поддавался оценкам, как качественным, так и количественным и т.п.

- Часто эти ошибки были следствием неполноты функциональных спецификаций или несогласованности между спецификациями и результатами проектирования.
- Вскоре был выдвинут тезис: совершенствование методов анализа есть ключ к созданию систем, эффективных по стоимости, производительности и надежности. Для решения ключевых проблем традиционного создания систем широкого профиля требовались новые методы, специально предназначенные для использования на ранних стадиях процесса

- Применение SADT проистекало из этого убеждения.
- Методы, подобные SADT, на начальных этапах создания системы позволяли гораздо лучше понять рассматриваемую проблему. А это сокращает затраты как на создание, так и на эксплуатацию системы, а кроме того, повышает ее надежность

- SADT - это способ уменьшить количество дорогостоящих ошибок за счет структуризации на ранних этапах создания системы, улучшения контактов между пользователями и разработчиками и сглаживания перехода от анализа к проектированию.

- Дуглас Т. Росс часть своих PLEX-теорий относящихся к методологии и языку описания систем, назвал "Методология структурного анализа и проектирования" (SADT). Исходная работа над SADT началась в 1969 г. Первое ее крупное приложение было реализовано в 1973 г. при разработке большого аэрокосмического проекта, когда она была несколько пересмотрена сотрудниками SofTech, Inc



- Появление SADT на рынке произошло в 1975 г. после годовичного оформления в виде продукта.
- К 1981 г. SADT уже использовали более чем в 50 компаниях при работе более чем над 200 проектами, включавшими более 2000 людей и охватывавшими дюжину проблемных областей, в том числе телефонные сети, аэрокосмическое производство, управление и контроль, учет материально-технических ресурсов и обработку данных.

# SADT-модели

- Описание системы с помощью SADT называется моделью.
- В SADT-моделях используются как естественный, так и графический языки.
- Для передачи информации о конкретной системе источником естественного языка служат люди, описывающие систему, а источником графического языка - сама методология SADT

- С точки зрения SADT модель может быть сосредоточена либо на функциях системы, либо на ее объектах.
- SADT-модели, ориентированные на функции, принято называть функциональными моделями (IDEF0), а ориентированные на объекты системы - моделями данных (IDEF1x).

- функциональная модель представляет с требуемой степенью детализации систему функций, которые в свою очередь отражают свои взаимоотношения через объекты системы.
- Модели данных дуальны к функциональным моделям и представляют собой подробное описание объектов системы, связанных системными функциями.
- Полная методология SADT поддерживает создание множества моделей для более точного описания сложной системы

# Модель отвечает на вопросы

- SADT-модель дает полное, точное и адекватное описание системы, имеющее конкретное назначение. Это назначение, называемое целью модели, вытекает из формального определения модели в SADT:
- $M$  есть модель системы  $S$ , если  $M$  может быть использована для получения ответов на вопросы относительно  $S$  с точностью  $A$ .

- Таким образом, целью модели является получение ответов на некоторую совокупность вопросов.
- Эти вопросы неявно присутствуют (подразумеваются) в процессе анализа и, следовательно, они руководят созданием модели и направляют его.
- Это означает, что сама модель должна будет дать ответы на эти вопросы с заданной степенью точности.
- Если модель отвечает не на все вопросы или ее ответы недостаточно точны, то мы говорим, что модель не достигла своей цели.

# . Модель имеет единственный субъект

- Модель является некоторым толкованием системы. Поэтому субъектом моделирования служит сама система. Однако моделируемая система никогда не существует изолированно: она всегда связана с окружающей средой. Причем зачастую трудно сказать, где кончается система и начинается среда. По этой причине в методологии SADT подчеркивается необходимость точного определения границ системы. SADT-модель всегда ограничивает свой субъект, т.е. модель устанавливает точно, что является и что не является субъектом моделирования, описывая то, что входит в систему, и подразумевая то, что лежит за ее пределами.

# У модели может быть только одна точка зрения

- С определением модели тесно связана позиция, с которой наблюдается система и создается ее модель. Поскольку качество описания системы резко снижается, если оно не сфокусировано ни на чем, SADT требует, чтобы модель рассматривалась все время с одной и той же позиции. Эта позиция называется "точкой зрения" данной модели



- "Точку зрения" лучше всего представлять себе как место (позицию) человека или объекта, в которое надо встать, чтобы увидеть систему в действии

# Модели как взаимосвязанные наборы диаграмм

- После того как определены субъект, цель и точка зрения модели, начинается первая интеграция процесса моделирования по методологии SADT.
- Субъект определяет, что включить в модель, а что исключить из нее. Точка зрения диктует автору модели выбор нужной информации о субъекте и форму ее подачи.
- Цель становится критерием окончания моделирования.
- Конечным результатом этого процесса является набор тщательно взаимоувязанных описаний, начиная с описания самого верхнего уровня всей системы и кончая подробным описанием деталей или операций системы

- Каждое из таких тщательно взаимосогласованных описаний называется диаграммой.
- SADT-модель объединяет и организует диаграммы в иерархические структуры, в которых диаграммы наверху модели менее детализированы, чем диаграммы нижних уровней.

- Сложности, связанные с описанием многих искусственных систем, объясняются тем, что эти системы слишком велики для того, чтобы можно было просто перечислить все их компоненты.
- С другой стороны, они могут быть упрощены за счет обобщающих предположений.
- Методология SADT создана специально для представления таких сложных систем путем построения моделей

- SADT-модель - это описание системы, у которого есть единственный субъект, цель и одна точка зрения.
- Целью служит набор вопросов, на которые должна ответить модель.
- Точка зрения - позиция, с которой описывается система.
- Цель и точка зрения - это основополагающие понятия SADT

- Описание модели SADT организовано в виде иерархии взаимосвязанных диаграмм.
- Вершина этой древовидной структуры представляет собой самое общее описание системы, а ее основание состоит из наиболее детализированных описаний.

# Синтаксис и применение диаграмм

- Диаграмма является основным рабочим элементом при создании модели.
- Разработчик диаграмм и моделей обычно называется аналитиком, или, в терминологии SADT, автором.
- Диаграммы имеют собственные синтаксические правила, отличающиеся от синтаксических правил моделей

# Диаграммы содержат блоки и дуги

- Каждая SADT-диаграмма содержит блоки и дуги.
- Блоки изображают функции моделируемой системы.
- Дуги связывают блоки вместе и отображают взаимодействия и взаимосвязи между ними.
- Диаграмме дается название, которое располагается в центре нижней части ее бланка.
- На каждой диаграмме написана стандартно идентифицирующая ее информация: автор диаграммы, частью какого проекта является работа, дата создания или последнего пересмотра диаграммы, статус диаграммы.
- Вся идентифицирующая информация располагается в верхней части бланка диаграммы



# Блоки представляют функции

- Функциональные блоки на диаграммах изображаются прямоугольниками.
- Блок представляет функцию или активную часть системы, поэтому названиями блоков служат глаголы или глагольные обороты. Например, названиями блоков диаграммы выполнить задание являются: определить степень выполнения задания, выбрать инструменты, подготовить рабочее место, обработать на станке и собрать.

- Кроме того, SADT требует, чтобы в диаграмме было не менее трех и не более шести блоков. Эти ограничения поддерживают сложность диаграмм и модели на уровне, доступном для чтения, понимания и использования. Другими словами, SADT-диаграммы и SADT-модели наглядны.

- В отличие от других графических методов структурного анализа в SADT каждая сторона блока имеет особое, вполне определенное назначение.
- Левая сторона блока предназначена для ВХОДОВ,
- верхняя - для управления,
- правая - для ВЫХОДОВ,
- нижняя - для механизмов.
- Такое обозначение отражает определенные системные принципы: входы преобразуются в выходы, управление ограничивает или предписывает условия выполнения преобразований, механизмы показывают, кто, что и как выполняет функция.

# Блоки имеют доминирование

- Блоки SADT никогда не размещаются на диаграмме случайным образом. Они размещаются по степени важности, как ее понимает автор диаграммы. В SADT этот относительный порядок называется доминированием.
- Доминирование понимается как влияние, которое один блок оказывает на другие блоки диаграммы.
- Например, самым доминирующим блоком диаграммы может быть либо первый из требуемой последовательности функций, либо планирующая или контролирующая функция, влияющая на все

- Наиболее доминирующий блок обычно размещается в верхнем левом углу диаграммы, а наименее доминирующий - в правом нижнем углу.
- В результате получается "ступенчатая" схема.
- Расположение блоков на странице отражает авторское определение доминирования. Таким образом, топология диаграммы показывает, какие функции оказывают большее влияние на остальные

- Блоки в SADT должны быть перенумерованы. Номера блоков служат однозначными идентификаторами для системных функций и автоматически организуют эти функции в иерархию модели. Используя номера блоков и оценивая влияние, которое один блок оказывает на другой;
- аналитик может организовать модель по принципу функционального доминирования. Это позволяет согласовать иерархический порядок функций в модели с уровнем влияния каждой функции на остальную часть системы. Поэтому настоятельно рекомендуется по мере возможности нумеровать блоки в соответствии с порядком их доминирования.

# Дуги изображают объекты

- Дуги на SADT-диаграмме изображаются одинарными линиями со стрелками на концах.
- Для функциональных SADT-диаграмм дуга представляет множество объектов
- Так как в SADT дуги изображают объекты, они описываются (помечаются) существительными или существительными с определениями, располагающимися достаточно близко к линии дуги

# Дуги изображают взаимосвязи между блоками

- Между объектами и функциями возможны четыре отношения: вход, управление, выход, механизм.
- Каждое из этих отношений изображается дугой, связанной с определенной стороной блока.
- По соглашению левая сторона блока предназначена для входных дуг, верхняя сторона - для управленческих дуг, правая сторона - для выходных дуг, нижняя сторона - для дуг механизмов.
- Таким образом, стороны блока чисто графически сортируют объекты, изображаемые касающимися блока дугами



- Входные дуги изображают объекты, используемые и преобразуемые функциями. Например, в процессе изготовления детали *сырье* трансформируется функцией *обработать на станке и собрать*.
- Управленческие дуги представляют информацию, управляющую действиями функций. Обычно управляющие дуги несут информацию, которая указывает, как должна выполняться функция. Например, *техническая карта задания* определяет, какие нужно выбрать инструменты, какие потребуются станки и цеха и как инструменты и станки должны использоваться при изготовлении детали

- Выходные дуги изображают объекты, в которые преобразуются входы. Например, *обработать на станке и собрать* преобразует *сырье и брак* в *результаты обработки*, которые в конечном итоге становятся деталями.
- Дуги механизмов отражают, по крайней мере частично, как функции (т.е. функции системы) реализуются. Таким образом, *механизмы* изображают физические аспекты функции (склады, люди, организации, приборы)

- Итак, SADT-диаграмма составлена из блоков, связанных дугами, которые определяют, как блоки влияют друг на друга. Это влияние может выражаться либо в передаче выходной информации к другой функции для дальнейшего преобразования, либо в выработке управляющей информации, предписывающей, что именно должна выполнять другая функция

- Таким образом, SADT-диаграммы не являются ни блок-схемами, ни просто диаграммами потоков данных. Это предписывающие диаграммы, представляющие входные-выходные преобразования и указывающие правила этих преобразований. Дуги на них изображают интерфейсы между функциями системы, а также между системой и ее окружающей средой

# Типы взаимосвязей между блоками

- В методологии SADT требуется только пять типов взаимосвязей между блоками для описания их отношений: управление, вход, обратная связь по управлению, обратная связь по входу, выход-механизм.
- Связи по управлению и входу являются простейшими, поскольку они отражают прямые воздействия, которые интуитивно понятны и очень просты. Отношение управления возникает тогда, когда выход одного блока непосредственно влияет на блок с меньшим доминированием

- Отношение входа возникает тогда, когда выход одного блока становится входом для блока с меньшим доминированием,
- Обратная связь по управлению и обратная связь по входу являются более сложными, поскольку они представляют итерацию или рекурсию. А именно выходы из одной функции влияют на будущее выполнение других функций, что впоследствии влияет на исходную функцию. Обратная связь по управлению возникает тогда, когда выход некоторого блока влияет на блок с большим доминированием
- Связь по входной обратной связи имеет место тогда, когда выход одного блока становится входом другого блока с большим доминированием. Например, задания, отвергнутые функцией *контролировать качество выполнения*, отсылаются на вход блока *выполнить задание* в качестве брака

- Связи "выход-механизм" встречаются нечасто и представляют особый интерес. Они отражают ситуацию, при которой выход одной функции становится средством достижения цели для другой

# Дуги представляют наборы объектов

- Дуга в SADT редко изображает один объект. Обычно она символизирует набор объектов. Например, дуга, именуемая *рабочий комплект*, отражает *техническое задание*, *чертеж*, *план-график*, *некоторое сырье и заготовки*



- Так как дуги представляют наборы объектов, они могут иметь множество начальных точек (источников) и конечных точек (назначений). Поэтому дуги могут разветвляться и соединяться различными сложными способами. Вся дуга или ее часть может выходить из одного или нескольких блоков и заканчиваться в одном или нескольких блоках

- Для объяснения того, как дуги представляют разъединение и соединение наборов объектов, в SADT были разработаны специальные соглашения относительно представления и описания разветвлений и соединений дуг.

# Разветвление дуг

- Разветвления дуг, изображаемые в виде расходящихся линий, означают, что все содержимое дуг или его часть может появиться в каждом ответвлении дуги. Дуга всегда помечается до разветвления, чтобы дать название всему набору. Кроме того, каждая ветвь дуги может быть помечена или не помечена в соответствии со следующими правилами
- непомеченные ветви содержат все объекты, указанные в метке дуги перед разветвлением (т.е. все объекты принадлежат этим ветвям);
- ветви, помеченные после точки разветвления, содержат все объекты или их часть, указанные в метке дуги перед разветвлением (т.е. каждая метка ветви уточняет, что именно содержит ветвь).

# Слияние дуг

- Слияние дуг в SADT, изображаемое как сходящиеся вместе линии, указывает, что содержимое каждой ветви идет на формирование метки для дуги, являющейся результатом слияния исходных дуг.
- После слияния результирующая дуга всегда помечается для указания нового набора объектов, возникшего после объединения. Кроме того, каждая ветвь перед слиянием может помечаться или не помечаться в соответствии со следующими правилами:
- непомеченные ветви содержат все объекты, указанные в общей метке дуги после слияния (т.е. все объекты исходят из всех ветвей);
- помеченные перед слиянием ветви содержат все или некоторые объекты из перечисленных в общей метке после слияния (т.е. метка ветви ясно указывает, что содержит ветвь).

# Идентификация версий диаграмм С-номерами

- При создании SADT-модели одну и ту же диаграмму вместе с ее блоками и дугами перечерчивают несколько раз, что приводит к появлению различных ее вариантов. Чтобы различать разные версии одной и той же диаграммы, в SADT используется схема контроля конфигурации диаграмм, основанная на хронологических номерах, или С-номерами.
- С-номерные коды образуются из инициалов автора и последовательных номеров. Эти коды ставятся в нижнем правом углу SADT-бланка

- Если диаграмма заменяет более старый вариант, то автор помещает предыдущий S-номер в скобках, чтобы указать на связь с предыдущей работой. Например, диаграмма DAM010 заменяет предыдущую версию DAM009.
- Каждый автор проекта SADT ведет реестр всех созданных им диаграмм, нумеруя их последовательными целыми числами

# Резюме

- SADT-диаграмма содержит от трех до шести блоков, связанных дугами, и имеет при построении модели несколько версий.
- Для того чтобы различать версии одной и той же диаграммы, используются С-номера.
- Блоки на диаграмме изображают системные функции, а дуги изображают множество различных объектов системы.
- Блоки обычно располагаются на диаграмме в соответствии с порядком их доминирования, т.е. их важностью относительно друг друга.
- Дуги, связывающие блоки, изображают наборы объектов и могут разветвляться и соединяться различными сложными способами.
- Однако, разветвляясь и соединяясь, дуги должны во всех случаях сохранять представляемые ими объекты.

# Синтаксис моделей и работа с НИМИ

- Одна SADT-диаграмма сложна сама по себе, поскольку она содержит от трех до шести блоков, связанных множеством дуг. Для адекватного описания системы требуется несколько таких диаграмм.
- Диаграммы, собранные и связанные вместе, становятся SADT-моделью. В SADT дополнительно к правилам синтаксиса диаграмм существуют правила синтаксиса моделей.
- Синтаксис SADT-моделей позволяет аналитику определить границу модели, связать диаграммы в одно целое и обеспечить точное согласование между диаграммами.
- Никакой другой метод структурного анализа не позволяет так точно, как SADT, соединять диаграммы в тщательно организованные комплекты, называемые моделями



# Система представляется одним блоком

- SADT-модель является иерархически организованной совокупностью диаграмм.
- Диаграммы обычно состоят из трех-шести блоков, каждый из которых потенциально может быть детализирован на другой диаграмме.
- Каждый блок может пониматься как отдельный тщательно определенный объект.
- Разделение такого объекта на его структурные части (блоки и дуги, составляющие диаграмму) называется декомпозицией

- Декомпозиция формирует границы, и каждый блок в SADT рассматривается как формальная граница некоторой части целой системы, которая описывается. Другими словами, блок и касающиеся его дуги определяют точную границу диаграммы, представляющей декомпозицию этого блока.
- Эта диаграмма, называемая диаграммой с потомком, описывает все, связанное с этим блоком и его дугами, и не описывает ничего вне этой границы.
- Декомпозируемый блок называется родительским блоком, а содержащая его диаграмма - соответственно родительской диаграммой. Таким образом SADT-диаграмма является декомпозицией некоторого ограниченного объекта.

- Принцип ограничения объекта встречается на каждом уровне.
- Один блок и несколько дуг на самом верхнем уровне используются для определения границы всей системы. Этот блок описывает общую функцию, выполняемую системой.
- Дуги, касающиеся этого блока, описывают главные управления, входы, выходы и механизмы этой системы.
- Диаграмма, состоящая из одного блока и его дуг, определяет границу системы и называется контекстной диаграммой модели. Таким образом, этот блок изображает границу системы: все, лежащее внутри него, является частью описываемой системы, а все, лежащее вне него, образует среду системы.

# . Идентификация декомпозиции номерами узлов

- SADT-модели развиваются в процессе структурной декомпозиции сверху вниз.
- Сначала декомпозируется один блок, являющийся границей модели, на одной диаграмме, которая имеет от трех до шести блоков, затем декомпозируется один (или больше) из этих блоков на другой диаграмме с тремя-шестью блоками и т.д.
- Название диаграммы совпадает с названием декомпозируемого блока. Результатом этого процесса является модель, диаграмма верхнего уровня которой описывает систему в общих терминах "черного ящика", а диаграммы нижнего уровня описывают очень детализированные аспекты и операции системы.

- Таким образом, каждая диаграмма представляет собой некоторую законченную часть всей модели.
- В методологии SADT идентифицируется каждая диаграмма данной модели по средством того, что называется "номер узла".
- Номер узла для контекстной диаграммы имеет следующий вид: название модели или аббревиатура, косая черта, заглавная буква A (Activity в функциональных диаграммах), дефис и ноль.

- Например, номером узла для контекстной диаграммы модели экспериментального механического цеха является ЭМЦ/А-0.
- Номером узла диаграммы, декомпозирующей контекстную диаграмму, является тот же номер узла, но без дефиса (например, ЭМЦ/АО).
- Все другие номера узлов образуются посредством добавления к номеру узла родительской диаграммы номера декомпозируемого блока. Номер узла на первой диаграмме - ЭМЦ/АО, а номер узла на второй диаграмме - ЭМЦ/А1. Диаграмма ЭМЦ/А1 декомпозирует блок 1 диаграммы ЭМЦ/АО. (Первый ноль при образовании номера узла принято опускать, поэтому вместо ЭМЦ/АО1 пишется ЭМЦ/А1.)

# . Связывание декомпозиции с помощью C-номеров

- Помимо использования для идентификации версий диаграмм, C-номера применяются для связки диаграмм при движении как вверх, так и вниз по иерархии модели. Обычно C-номер диаграммы, декомпозирующей некоторый блок, впервые появляется непосредственно под этим блоком на родительской диаграмме. Это образует "направленную вниз" связь от родительской диаграммы к диаграмме-потомку

- Как только образуется направленная вниз связь, на диаграмме-потомке формируется ссылка на родительскую диаграмму.
- В области контекста SADT-бланка (правый верхний угол) автор изображает каждый блок родительской диаграммы маленькими квадратиками, заштриховывает квадратик декомпозируемого блока и размещает С-номер родительской диаграммы возле заштрихованного квадратика.
- Это образует "направленную вверх" (к родительской диаграмме) связь. Метод соединения диаграмм посредством однозначно определенных номеров гарантирует, что именно нужная версия диаграммы станет частью модели. Другими словами, при использовании С-номеров осуществляется тщательный контроль за введением новых диаграмм в иерархию модели.



# Коды ISOM гарантируют стыковку диаграмм

- Хорошая методология структурного анализа, позволяющая создавать отдельные диаграммы, должна гарантировать правильное соединение всех диаграмм для образования согласованной модели.
- SADT-диаграммы имеют внешние дуги -дуги, как бы выходящие наружу и ведущие к краю страницы. Эти дуги являются интерфейсом между диаграммой и остальной частью модели. SADT требует, чтобы все внешние дуги диаграммы были согласованы с дугами, образующими границу этой диаграммы
- Другими словами, диаграмма должна быть "состыкована" со своей родительской диаграммой. Обычно это означает, что внешние дуги согласованы по числу и наименованию (но не обязательно по расположению) с дугами, касающимися декомпозированного блока родительской диаграммы.

- В SADT принята система обозначений, позволяющая аналитику точно идентифицировать и проверять связи по дугам между диаграммами.
- Эта схема кодирования дуг -"ICOM" - получила название по первым буквам английских эквивалентов слов вход (Input), управление (Control), выход (Output), механизм (Mechanism).
- Коды ICOM чрезвычайно эффективны, поскольку они позволяют аналитику быстро проверять согласованность внешних дуг диаграммы с граничными дугами соответствующего блока родительской диаграммы. Они также обеспечивают согласованность декомпозиции, поскольку все дуги, входящие в диаграмму и выходящие из нее, должны быть учтены.

- Если вы начинаете строить диаграмму следующего уровня, то дуги, касающиеся декомпозируемого блока, используются в качестве источников и приемников для дуг, которые вы создаете на новой диаграмме.
- После завершения диаграммы ее внешние дуги стыкуются с родительской диаграммой для обеспечения согласованности

Одним из способов такой стыковки может служить присваивание кодов ISOM внешним дугам новой диаграммы согласно следующим правилам:

- представьте себе рисунок новой диаграммы внутри разлагаемого блока. Продлите внешние дуги почти до края диаграммы. Зрительно соедините каждую внешнюю дугу диаграммы с соответствующей граничной дугой декомпозируемого блока.
- присвойте код каждой зрительной связи. Используйте I для входных дуг, C - для связей между дугами управления, O - для связей между выходными дугами, M - для связей между дугами механизма.
- добавьте после каждой буквы цифру, соответствующую положению данной дуги среди других дуг того же типа, касающихся родительского блока. Причем входные и выходные дуги пересчитываются сверху вниз, а дуги управлений и механизмов пересчитываются слева направо. Теперь запишите каждый код около окончания каждой внешней дуги.

- При следовании схеме кодирования ISOM создается совокупность неявных связующих звеньев между страницами, которые можно быстро изменить при изменении границ .
- Эти неявные межстраничные связующие звенья облегчают процесс чтения и рецензирования SADT-диаграмм, а также проверку, насколько согласованно произведена декомпозиция.
- Коды ISOM упрощают также работу, связанную с внесением вручную локальных изменений в диаграмму, и объединяют различные варианты диаграмм так, что они хорошо стыкуются в модели

# Дуги "входят в тоннель"

- . Дуга "входит в тоннель", либо (1) если она является внешней дугой, которая отсутствует на родительской диаграмме (имеет скрытый источник), либо (2) если она касается блока, но не появляется на диаграмме, которая его декомпозирует (имеет скрытый приемник).
- Тоннельные дуги от скрытого источника начинаются скобками, чтобы указать, что эти дуги идут из какой-то другой части модели или прямо извне модели.
- Тоннельные дуги, имеющие скрытый приемник, кончаются скобками, чтобы отразить тот факт, что такая дуга идет к какой-то другой части модели или выходит из нее или что она не будет более в этой модели рассматриваться

# Резюме

- SADT-диаграммы являются декомпозициями ограниченных объектов. Объект ограничивается блоком и касающимися его дугами. Диаграмма, содержащая границу, называется родительской диаграммой, а диаграмма, декомпозирующая блок родительской диаграммы, называется диаграммой-потомком. Для связывания родительской диаграммы и диаграммы-потомка используются С-номера, так что модель всегда сохраняет актуальность. Коды ISOM используются для того, чтобы стыковать диаграмму-потомка с родительской диаграммой. Номер узла идентифицирует уровень данной диаграммы в иерархии модели. Когда диаграммы в модели становятся слишком трудными для чтения, для упрощения описания системы могут разумным образом использоваться специальные технические приемы типа "вхождения дуг в тоннель"

# Процесс моделирования

- В значительной мере успех методологии SADT объясняется ее графическим языком, хотя не менее ценным является сам процесс моделирования.
- Процесс моделирования в SADT включает сбор информации об исследуемой области, документирование полученной информации и представление ее в виде модели и уточнение модели посредством итеративного рецензирования.
- Кроме того, этот процесс подсказывает вполне определенный путь выполнения согласованной и достоверной структурной декомпозиции, что является ключевым моментом в квалифицированном анализе системы. SADT уникальна в своей способности обеспечить как графический язык, так и процесс создания непротиворечивой и полезной системы описаний



- SADT является методологией в полном смысле, потому что она объединяет итеративный процесс создания модели, нотации, управляющие конфигурацией модели, язык ссылок для диаграмм, язык функций моделей с графическим языком описания системы, а также рекомендации по реализации аналитических проектов.
- Нотации, управляющие конфигурацией, гарантируют, что новые диаграммы будут корректно встроены в иерархическую структуру модели.
- Язык ссылок в SADT, правила сокращений для ссылок, адресованных к отдельным частям диаграммы, облегчают оформление замечаний при рецензировании модели.
- Язык функций позволяет декларативно определять правила работы системы, что часто является особенно важным завершающим шагом в описании системы