# **Методология SADT и стандарты IDEF**

Методология SADT - одна из самых известных методологий анализа и проектирования систем. Она является, пожалуй, единственной методологий, отражающей такие характеристики, как управление, обратная связь и ресурсы. Другая особенность SADT заключается в том, что она развивалась как язык описания функционирования систем общего вида, тогда как в других структурных методологиях упор чаще делается на проектирование программного обеспечения.

Автор методологии, Дуглас Росс, в 1969 г. часть своих теорий, относящихся к методологии и языку описания систем, назвал SADT "Structured Analysis and Design Technique" ("Методология структурного анализа и проектирования"). Первое ее крупное приложение было реализовано в 1973 г. при разработке большого аэрокосмического проекта, а на рынке SADT появляется в 1975 г.

Описание системы с помощью SADT называется моделью, при этом используются как естественный, так и графические языки. SADT-модель может быть сосредоточена либо на функциях системы, либо на ее объектах. Модели, ориентированные на функции, принято называть функциональными, а на объекты системы моделями данных.

С помощью SADT-методологии решаются следующие основные задачи (для систем любой природы):

анализ функций, выполняемых системой;

описание спецификаций требований и функций проектируемой системы;

проектирование системы.

Более 10 лет SADT была "бумажной" технологией, но в середине 80-х годов, когда появились персональные компьютеры с графическими возможностями, SADT "пересела" за компьютер. Одним из первых программных комплексов структурно-функционального анализа на основе SADT был пакет AUTOIDEF, разработанный в рамках программы ВВС США по созданию интегрированной автоматизированной системы управления производством. В основе пакета лежит доведенное до уровня стандарта подмножество SADT методология IDEF, состоящая из трех методологий:

IDEF0 функциональное моделирование;

IDEF1 информационное моделирование;

IDEF2 динамическое моделирование функций, информации и ресурсов.

Методология IDEF, основанная на принципах системного анализа и предназначенная для представления функций произвольной системы (будь то управление финансами, организация работ, обучение или автоматизация), фактически стала стандартом не только в США, но и в ряде европейских стран. Из трех названных методологий наибольшее распространение получила первая IDEF0. В 1985 г. методология IDEF1 была расширена и переименована в IDEF1X. Что-касается методологии IDEF2, то она не получила широкого распространения.

Основные средства ССА

Сегодня существует богатая палитра методологий и инструментальных средств ССА. Наиболее распространены следующие методологии:

SADT - методология структурного анализа и проектирования.

IDEF0 - методология функционального моделирования, являющаяся составной частью SADT и позволяющая описать бизнес-процесс в виде иерархической системы взаимосвязанных функций.

IDEF1X - методология информационного моделирования, являющаяся составной частью SADT и основанная на концепции "сущность связь".

IDEF3 - методология описания процессов, рассматривающая последовательность выполнения и причинно-следственные связи между ситуациями и событиями для структурного представления знаний о системе.

IDEF4 - методология объектно-ориентированного проектирования сложных систем, описывающая структуру, поведение и реализацию систем с использованием терминов класса объектов.

IDEF5 - методология онтологического анализа систем, т.е. анализа основных терминов и понятий (словаря), используемых для характеристики объектов и процессов, границ использования, взаимосвязей между ними.

DFD (Data Flow Diagrams - диаграммы потоков данных) методология структурного анализа, описывающая внешние по отношению к системе источники и адреса, логические функции, потоки и хранилища данных, к которым осуществляется доступ.

ERD (Entity-Relationship Diagrams - диаграммы "сущность - связь") способ определения данных и отношений между ними, обеспечивающий детализацию хранилищ данных проектируемой системы, включая идентификацию объектов (сущностей), свойств этих объектов (атрибутов) и их отношений с другими объектами (связей).

STD (State Transition Diagrams - диаграммы переходов состояний) методология моделирования последующего функционирования системы на основе ее предыдущего и текущего функционирования.

CRN (Color Petri Nets - раскрашенные сети Петри) методология создания динамической модели бизнес-процесса, позволяющая проанализировать зависящие от времени характеристики процесса и распределение ресурсов для входящих потоков различной структуры.

ABC (Activity Based Costing - функционально-стоимостный анализ) метод определения стоимости и других характеристик изделий и услуг на основе функций и ресурсов, задействованных в бизнес-процессах.

Используя перечисленные средства, можно создать полное описание экономической или информационной системы (того, что делает или должна делать система).

# **Функциональное моделирование**

Задача функционального моделирования состоит в представлении системы в виде совокупности взаимосвязанных функций. В качестве методологического инструмента функционального моделирования рассмотрим методологию IDEF0, которая включает в себя метод IDEF0, а также методы и процедуры, его поддерживающие.

В методе IDEF0 можно выделить такие составляющие, как концепция метода, графический язык, процедура чтения диаграммы, метод построения модели, критерии оценки качества и др.

В структуру организационной поддержки метода IDEFO входят:

процедура сбора данных (интервьюирование);

метод групповой работы;

формы документирования модели;

процедуры согласования и утверждения модели.

Концепция IDEFO-моделей

IDEF0-модель описывает: что система делает, что она производит, какая информация используется для управления, какие ресурсы и средства применяются для исполнения ее функций.

Одним из достоинств IDEF0-моделей является то, что они обеспечивают возможность обмена информацией о рассматриваемом объекте на языке, понятном не только аналитику и разработчику системы, но и специалисту-эксперту в предметной области, пользователю, руководителю (Д. Росс назвал технику структурного анализа языком для передачи понимания). В основе метода IDEF0 лежат следующие концептуальные положения:

графическое представление модели в виде иерархии диаграмм, обеспечивающее компактность представления информации;

максимальная выразительность, т.е. способность наилучшим образом обеспечить "понимаемость" модели;

строгость и точность представления;

пошаговые процедуры разработки модели, ее просмотра и объединения;

отделение организации от функции исключение влияния организационной структуры на функциональную модель.

IDEFO-модель составляется из иерархического ряда диаграмм, которые постепенно отображают уровни все более подробных описаний функций и их интерфейсов в пределах системы. Диаграмма, находящаяся на вершине модели, обобщает всю рассматриваемую систему. Диаграммы первого уровня представляют важнейшие подсистемы с их взаимосвязями, а диаграммы самого нижнего уровня представляют детализированные функции, с помощью которых, собственно, и работает система.

Можно назвать три основных типа диаграмм, используемых в IDEF0-моделях: графические, текстовые и глоссарии.

Графические диаграммы главный компонент модели определяют функции и функциональные отношения. Эти функции в дальнейшем разбиваются (декомпозируются) на более детальные диаграммы, пока подсистема не будет описана на уровне, удовлетворяющем цели проекта.

Текстовые диаграммы и глоссарии (словари) обеспечивают дополнительную информацию для графических диаграмм. Кроме того, могут использоваться и так называемые поясняющие диаграммы FEO.

Однако до начала построения модели необходимо определиться с целью моделирования, границей системы и точкой зрения модели.

# **Цель моделирования**

Создаваемая IDEF0-модель имеет конкретное назначение, называемое целью модели. Цель моделирования можно определить с учетом следующего формального определения модели:

М есть модель системы S, если М может быть использована для получения ответов на вопросы относительно S с точностью А.

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". Если модель отвечает не на все вопросы или ее ответы недостаточно точны, то мы говорим, что модель "не достигла своей цели".

Обычно вопросы для IDEF0-модели формулируются на самом раннем этапе анализа или проектирования, при этом основная суть этих вопросов должна быть выражена в одной-двух фразах.

Пример. Определение цели модели работы деканата университета. Вопросы:

Каковы обязанности декана?

Каковы обязанности заместителя декана?

Каковы обязанности инспектора?

По каким вопросам работники деканата общаются со студентами?

По каким вопросам работники деканата общаются с кафедрами?

По каким вопросам работники деканата общаются с другими подразделениями университета?

Какая информация "приходит" в деканат и "уходит" из него?

Цель: Определить причины большой загруженности работников деканата.

Границы системы

Моделируемая система является частью окружающей среды и всегда связана с нею, поэтому зачастую трудно сказать, где кончается система и начинается среда. В связи с этим в методологии IDEF0 требуется определять границы системы. Модель устанавливает точно, что является и что не является объектом моделирования, описывая то, что входит в систему, и подразумевая то, что лежит за ее пределами. Как известно из системного анализа, границу системы можно указать, определив ее входы и выходы.

Точка зрения модели

С целью модели тесно связана позиция, с которой наблюдается система и создается ее модель. IDEF0 требует, чтобы конкретная модель всегда рассматривалась с одной и той же точки зрения. Эта позиция называется точкой зрения данной модели.

Пример. Определение точки зрения для модели деканата.

Претенденты: Декан. Заместители декана. Инспектор.

Выбор точки зрения: Поскольку основная работа деканата связана с учебным процессом, то наилучшей представляется точка зрения заместителя декана по учебной работе.

Точку зрения лучше всего представлять себе как "место" (позицию) человека или объекта, с которого можно "увидеть" систему в действии. В зависимости от цели моделирования, могут быть приняты различные положения точек зрения, что подчеркивает различные аспекты описания объекта. То, что является важным с одной точки зрения, может даже не появиться в модели, представленной с другой точки зрения для одной и той же системы. Для отражения в модели других точек зрения обычно используют-FЕО-диаграммы.

Синтаксис графических диаграмм

Компоненты синтаксиса IDEF0-диаграмм функциональные блоки и дуги (стрелки), правила и диаграммы. Функциональные блоки представляют функции, определенные как действия, процессы или преобразования. Дуги представляют данные или объекты, связанные с функциями. Правила определяют, как компоненты используются, а диаграммы служат инструментами для словесного или графического изображения моделей.

Функциональные блоки

Функциональный блок описывает то, что происходит в рассматриваемой части системы. Блок изображается в форме прямоугольника.

Он должен иметь название (имя) и номер внутри границ. Поскольку функциональный блок представляет функцию или активную часть системы, то его названием служит, глагол или - отглагольный - оборот. В настоящее время специалисты как бы разделились на два "лагеря": одни утверждают, что в названии блока следует употреблять неопределенную форму глагола, например, "Оформить командировку", другие считают допустимым использовать отглагольный оборот, например, "Оформление командировки".

Следует придерживаться следующих синтаксических правил оформления блоков:

выполняются сплошными линиями;

должны иметь прямоугольную форму, с прямыми углами;

должны быть достаточного размера, чтобы вставить название блока;

номер блока ставится внутри блока в нижнем правом углу.

Дуги

Дуга изображается одинарной линией со стрелкой на конце. Они изображают такие понятия, как данные или объекты, связанные с выполняемыми функциями, и описываются существительными или существительными с определениями.

Примеры наименований дуг: товары, платежи, законы, сотрудники, оборудование, командировочное задание, деньги.

Линия дуги может быть прямой или изогнутой. Поскольку дуга часто изображает не один, а несколько данных (объектов), то она может иметь разветвление или соединение.

Изображение дуг должно соответствовать следующим синтаксическим правилам:

могут быть изогнуты только на 90°;

изображаются сплошной линией;

чертятся только горизонтально или вертикально (но не по диагонали);

должны касаться внешней границы блока, но не должны входить в блок;

должны присоединяться к сторонам блока, но не к углам.

Взаимоотношения между дугами и блоками

Между данными (объектами) и функциями возможны четыре вида отношений: вход, управление, выход и механизм. Каждый вид изображается дугой, связанной с определенной стороной блока: левая сторона предназначена для входных дуг (входов) Х правая для выходных (выходов), верхняя сторона для управленческих дуг и нижняя для дуг механизмов.

Входные дуги изображают данные (объекты), используемые и преобразуемые функциями (документы, сырье, детали).

Выходные дуги изображают данные (объекты), в которые преобразуются входы (документы, счета, деньги, устройства).

Управляющие дуги представляют информацию, управляющую действиями функций (законы, приказы, системные требования, планы).

Дуги механизмов изображают физические аспекты функций (людей, склады, организации, приборы). С помощью дуг механизмов имеется возможность точно определять, какие ресурсы требуются для реализации конкретной функции, кто будет выполнять ее и т.д.

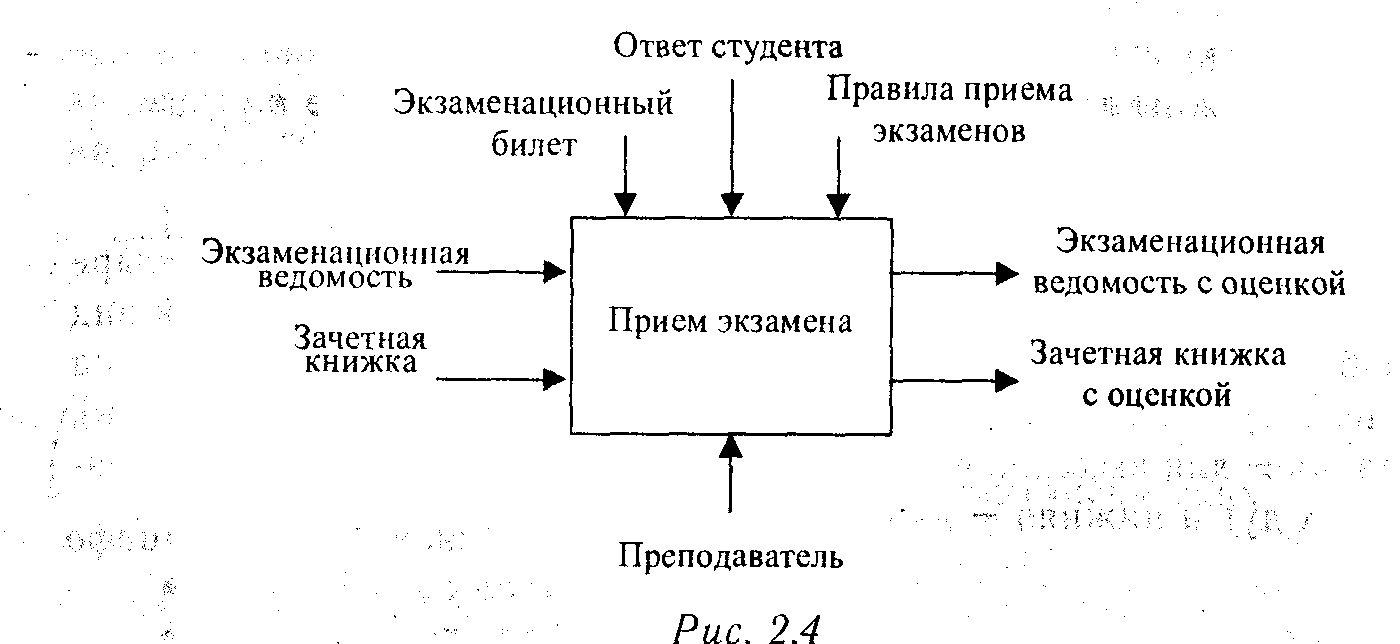
Пример. Процесс приема экзамена. Задача этой функции заключается в том, чтобы поставить оценки в экзаменационную ведомость и зачетную книжку. В качестве "механизма" здесь выступает преподаватель, который руководствуется содержанием экзаменационного билета, ответом студента и правилами приема экзаменов.

Нижняя сторона блока связана с еще одним типом дуг - дугой ссылки. Дуга ссылки (вызова) указывает подсистему, полностью выполняющую функцию данного блока. Это означает, что данный блок не имеет собственной детализирующей дочерней диаграммы, а детализирован полностью другим блоком в той же самой или другой модели. При этом множество вызывающих блоков могут вызывать один и тот же блок (по аналогии с программированием ссылку можно рассматривать как обращение к стандартной подпрограмме).

Стрелки ссылки должны направляться вниз и помечаться выражением ссылки на блок, который детализирует данный блок. Блок вызывающей диаграммы может вызывать только один блок. Однако, в зависимости от условий, указанных в примечании, приложенном к дуге запроса, блок вызывающей программы может выбирать одну из нескольких возможных называемых блоков. В этом случае, метка дуги ссылки должна включать список ссылок всех возможных называемых блоков.

Наименования (метки) дуг ставятся рядом со стрелкой. Если связь метки с соответствующей дугой не очевидна (для метки недостаточно места рядом с дугой), то для уточнения связи используют - ломаную линию.

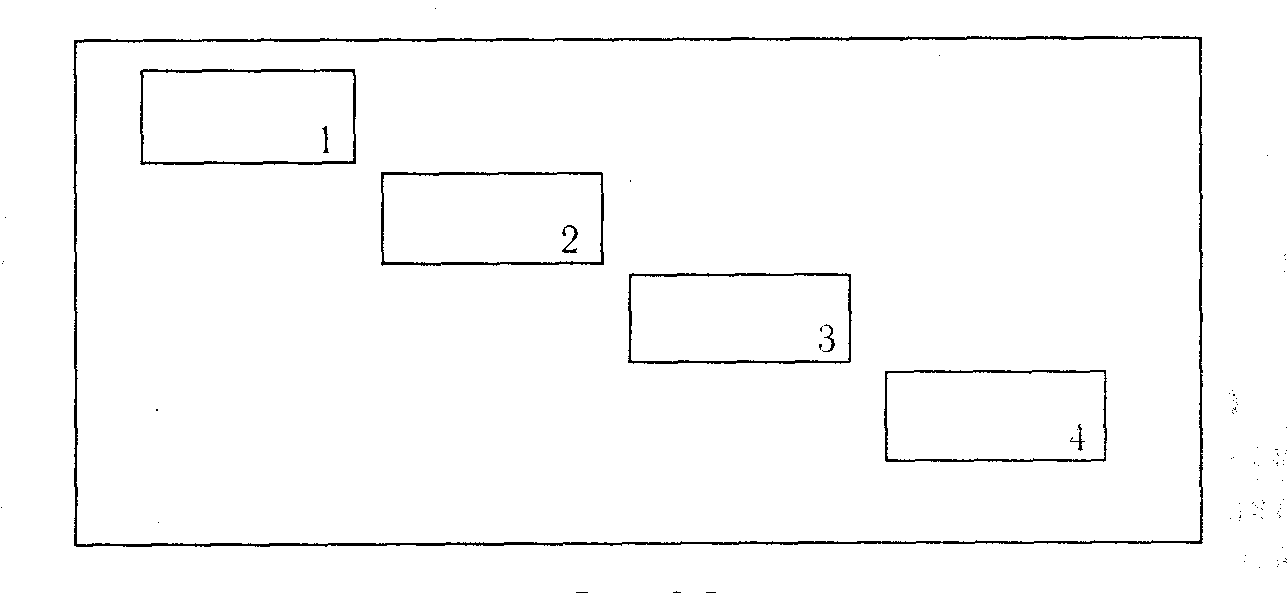
Входные дуги на диаграмме IDEF0 выступают как ограничения. Соединение выхода одного блока с входом, управлением или механизмом других показывает, что моделируемая функция требует (и таким образом ограничивается) присутствия соответствующего выхода предыдущего блока. Таким образом, входные дуги данного блока представляют все данные (объекты), которые необходимы для выполнения его функции.



Размещение блоков на диаграмме

На диаграмме блоки выстраиваются по степени важности (как это понимает автор!). Такой относительный порядок называется доминированием. Доминирование понимается как влияние одного блока диаграммы на другие. Наиболее доминирующий блок обычно размещается в верхнем левом углу диаграммы, а наименее доминирующий - в правом нижнем.

Другим методом указания доминирования блоков является их нумерация: блок с меньшим номером будет иметь большую степень доминирования над блоком с большим номером.

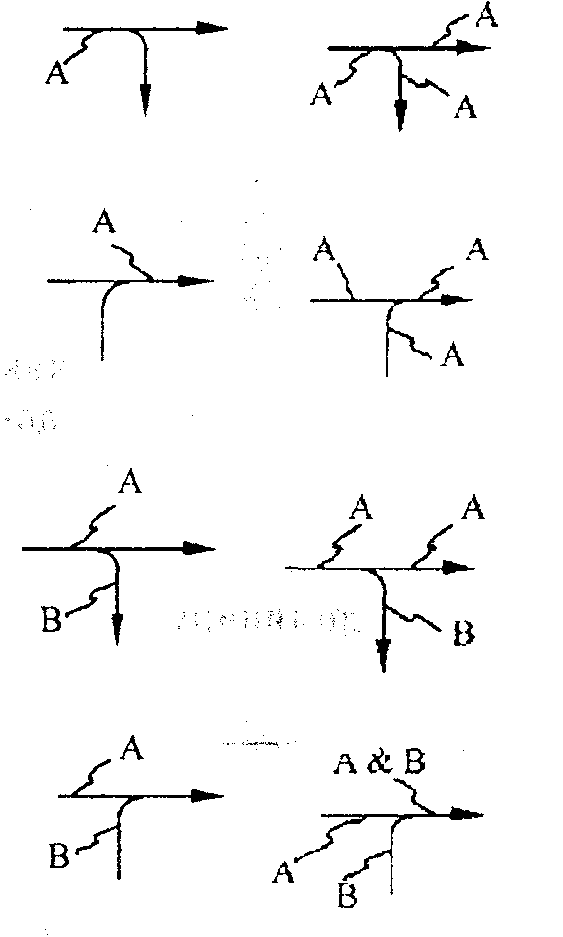


Разветвление и слияние дуг

Дуга в IDEF0 редко изображает один объект или одни данные. Обычно она отражает их набор, поэтому дуги могут разветвляться и соединяться различными сложными способами. Вся дуга или часть, ее может выходить из одного или нескольких блоков и заканчиваться в одном или нескольких блоках. Разветвление дуг, изображаемое в виде расходящихся линий, означает, что все содержимое дуг (или его часть) может появиться в каждом ответвлении дуги. При этом дуга помечается до ветвления, чтобы дать название всему набору. Кроме того, каждая ветвь дуги может быть помечена или не помечена в соответствии со следующими правилами:

непомеченные ветки содержат все данные (объекты), указанные в метке перед разветвлением;

ветки, помеченные после точки разветвления, содержат все данные (объекты) или их часть, указанные в метке, дуги перед разветвлением (т.е. каждая метка ветки уточняет, что именно содержит ветвь).



Слияние дуг, изображаемое в виде сходящихся вместе линий, 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 данных (объектов), возникшего после объединения. Кроме того, каждая ветвь перед слиянием может помечаться в соответствии со следующими правилами:

непомеченные ветки содержат все данные (объекты), указанные в обшей метке после слияния;

ветки, помеченные перед слиянием, содержат все данные (объекты) или их часть, перечисленные в метке дуги после слияния (т.е. каждая метка ветки ясно указывает, что именно содержит ветвь).

Связи между блоками

IDEF0-диаграмма составляется из блоков, связанных дугами, которые определяют, как блоки влияют друг на друга. Это влияние может выражаться либо в передаче результатов работы одного блока другому блоку для дальнейшего преобразования, либо в выработке управляющей информации, предписывающей, что именно должна выполнять другая функция. Можно выделить пять типов взаимосвязей между блоками для описания их отношений:

вход-управление;

выход-вход;

обратная связь по управлению;

обратная связь по входу;

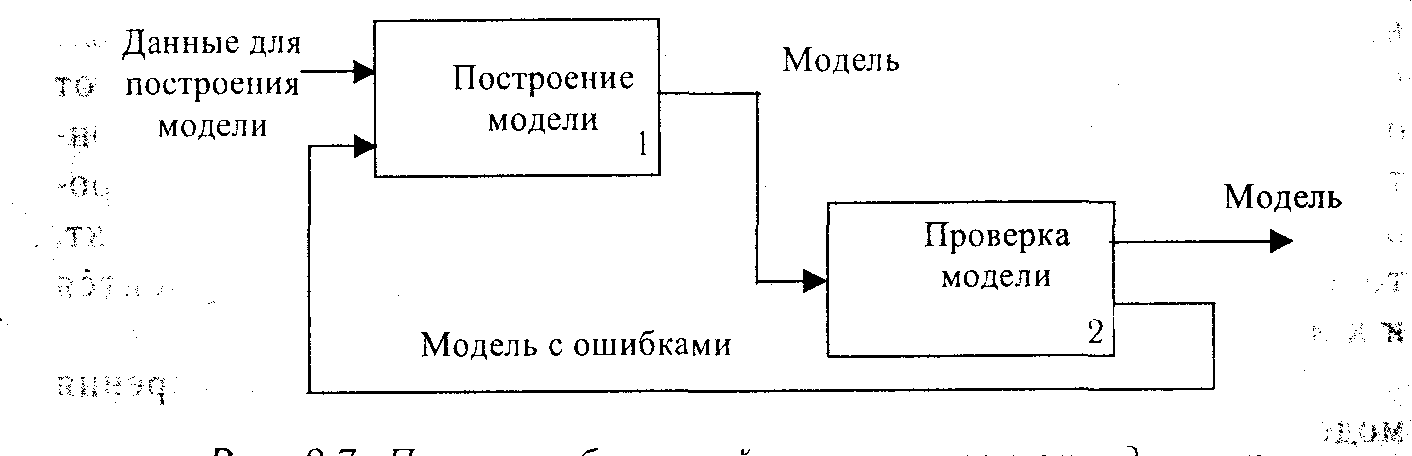
выход-механизм.

Отношение вход-управление возникает в том случае, если выход одного блока содержит управляющие данные для блока с меньшим доминированием.

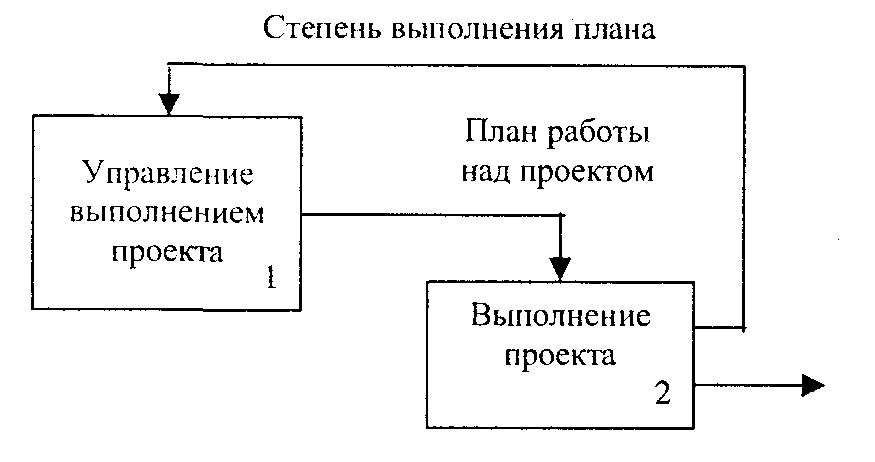
Отношение выход-вход возникает тогда, когда выход одного блока становится входом для блока с меньшим доминированием.

Более сложны обратные связи, поскольку они отражают итерационные процессы результаты работы функции (выходы) влияют на выполнение других функций, которые впоследствии влияют на исходную функцию. Различают описание двух видов обратной связи: по потоку данных (по входу) и по управлению.

Обратная связь по потоку данных возникает, когда выход одного блока становится входом другого блока с большим доминированием.



Управленческая обратная связь возникает, когда выход некоторого блока содержит управляющие данные для блока с большим доминированием.



Отношение выход-механизм встречается нечасто и отражают ситуацию, при которой выход одного блока становится средством достижения цели другого блока. Эти отношения характерны при распределении источников ресурсов (инструменты, обученный персонал, физическое пространство, оборудование, финансирование, материалы).