

IDEF0

IDEF0—FunctionModeling — методология функционального моделирования и графическая нотация, предназначенная для формализации и описания бизнес-процессов. Отличительной особенностью IDEF0 является её акцент на соподчинённость объектов. В IDEF0 рассматриваются логические отношения между работами, а не их временная последовательность.

В IDEF0 различают пять типов стрелок.

Вход - объекты, используемые и преобразуемые работой для получения результата (выхода). Допускается, что работа может не иметь ни одной стрелки входа. Стрелка входа рисуется как входящая в левую грань работы.

Управление - информация, управляющая действиями работы. Обычно управляющие стрелки несут информацию, которая указывает, что должна выполнять работа. Каждая работа должна иметь хотя бы одну стрелку управления, которая изображается как входящая в верхнюю грань работы.

Выход - объекты, в которые преобразуются входы. Каждая работа должна иметь хотя бы одну стрелку выхода, которая рисуется как исходящая из правой грани работы.

Механизм - ресурсы, выполняющие работу. Стрелка механизма рисуется как входящая в нижнюю грань работы. По усмотрению аналитика стрелки механизма могут не изображаться на модели.

Вызов - специальная стрелка, указывающая на другую модель работы. Стрелка вызова рисуется как исходящая из нижней части работы и используется для указания того, что некоторая работа выполняется за пределами моделируемой системы.

В методологии IDEF0 требуется только пять типов взаимодействий 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. Обратная связь по управлению и обратная связь по входу являются более сложными, поскольку представляют собой итерацию или рекурсию. А именно выходы из одной работы влияют на будущее выполнение других работ, что впоследствии повлияет на исходную работу.

Связи «выход-механизм» встречаются нечасто. Они отражают ситуацию, при которой выход одной функции становится средством достижения цели для другой.

Коэффициент несбалансированности:

$$K_b = \left| \frac{\sum_{i=1}^N A_i}{N} - \max_{i=1}^N (A_i) \right|$$

Необходимо стремиться, чтобы K_b был минимален для диаграммы.

DFD

Диаграммы потоков данных (DFD) - средство моделирования функциональных требований проектируемой системы. Разбивает требования на функциональные компоненты (процессы) и представляет их в виде сети, связанной потоками данных. Главная цель -

продемонстрировать, как каждый процесс преобразует свои входные данные в выходные, а также выявить отношения между этими процессами.

Для изображения DFD традиционно используются две различные нотации: Йодана (Yourdon) и Гейна-Сарсона (Gane-Sarson).



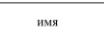
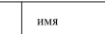


Компонента	Нотация Йодана	Нотация Гейна-Сарсона
ПОТОК ДАННЫХ	ИМЯ →	ИМЯ →
ПРОЦЕСС		
ХРАНИЛИЩЕ		
ВНЕШНЯЯ СУЩНОСТЬ		

Рисунок 1 – Основные компоненты диаграммы потоков данных

ПОТОКИ ДАННЫХ являются механизмами, использующимися для моделирования передачи информации (или даже физических компонент) из одной части системы в другую. Важность этого объекта очевидна: он дает название целому инструменту. Потоки на диаграммах обычно изображаются

именованными стрелками, ориентация которых указывает направление движения информации.

Назначение **ПРОЦЕССА** состоит в продуцировании выходных потоков из входных в соответствии с действием, задаваемым именем процесса. Это имя должно содержать глагол в неопределенной форме с последующим дополнением. Кроме того, каждый процесс должен иметь уникальный номер для ссылок на него внутри диаграммы. Этот номер может использоваться совместно с номером диаграммы для получения уникального индекса процесса во всей модели.

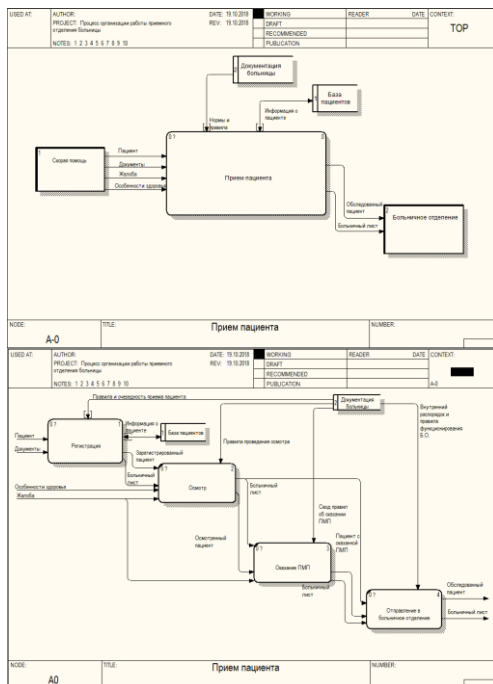
ХРАНИЛИЩЕ (НАКОПИТЕЛЬ)

ДАННЫХ позволяет на определенных участках определять данные, которые будут сохраняться в памяти между процессами. Фактически хранилище представляет "срезы" потоков данных во времени. Информация, которую оно содержит, может использоваться в любое время после ее определения, при этом данные могут выбираться в любом порядке. Имя хранилища должно идентифицировать его содержимое и быть существительным. В случае, когда поток данных входит или выходит в/из хранилища, и его структура соответствует структуре хранилища, он должен иметь то же самое имя, которое нет необходимости отражать на диаграмме.

ВНЕШНЯЯ СУЩНОСТЬ (или

ТЕРМИНАТОР) представляет сущность вне контекста системы, являющуюся источником или приемником системных данных. Ее имя должно содержать существительное. Предполагается, что объекты, представленные такими узлами, не должны участвовать ни в какой обработке.

Декомпозиция DFD осуществляется на основе процессов: каждый процесс может раскрываться с помощью DFD нижнего уровня.



Для обеспечения декомпозиции данных и некоторых других сервисных возможностей к DFD добавляются следующие типы объектов:

1) **ГРУППОВОЙ УЗЕЛ**. Предназначен для расщепления и объединения потоков. В некоторых случаях может отсутствовать (т.е. фактически вырождаться в точку слияния/расщепления потоков на диаграмме).

2) **УЗЕЛ-ПРЕДОК**. Позволяет увязывать входящие и выходящие потоки между детализируемым процессом и детализирующей DFD.

3) **НЕИСПОЛЬЗУЕМЫЙ УЗЕЛ**. Применяется в ситуации, когда декомпозиция данных производится в групповом узле, при этом требуются не все элементы входящего в узел потока.

4) **УЗЕЛ ИЗМЕНЕНИЯ ИМЕНИ**. Позволяет неоднозначно именовать потоки, при этом их содержимое эквивалентно. Например, если при проектировании разных частей системы один и тот же фрагмент данных получил различные имена, то эквивалентность соответствующих потоков данных обеспечивается узлом изменения имени. При этом один из потоков данных является входным для данного узла, а другой - выходным.

В соответствии с этими рекомендациями процесс построения модели разбивается на следующие этапы:

1) Расчленение множества требований и организация их в основные функциональные группы.

2) Идентификация внешних объектов, с которыми система должна быть связана.

3) Идентификация основных видов информации, циркулирующей между системой и внешними объектами.

4) Предварительная разработка контекстной диаграммы, на которой основные функциональные группы представляются процессами, внешние объекты - внешними сущностями, основные виды информации – потоками данных между процессами и внешними сущностями.

5) Изучение предварительной контекстной диаграммы и внесение в нее изменений по результатам ответов на возникающие при этом изучении вопросы по всем ее частям.

6) Построение контекстной диаграммы путем объединения всех процессов предварительной диаграммы в один процесс, а также группирования потоков.

7) Формирование DFD первого уровня на базе процессов предварительной контекстной диаграммы.

8) Проверка основных требований по DFD первого уровня.

9) Декомпозиция каждого процесса текущей DFD с помощью детализирующей диаграммы или спецификации процесса.

10) Проверка основных требований по DFD соответствующего уровня.

11) Добавление определений новых потоков в словарь данных при каждом их появлении на диаграммах.

12) Параллельное (с процессом декомпозиции) изучение требований (в том числе и вновь поступающих), разбиение их на элементарные и идентификация процессов или спецификаций процессов, соответствующих этим требованиям.

13) После построения двух-трех уровней проведение ревизии с целью проверки корректности и улучшения понимаемости модели.

14) Построение спецификации процесса (а не простейшей диаграммы) в случае, если некоторую функцию сложно или невозможно выразить комбинацией процессов.

7) Формирование DFD первого уровня на базе процессов предварительной контекстной диаграммы.

8) Проверка основных требований по DFD первого уровня.

9) Декомпозиция каждого процесса текущей DFD с помощью детализирующей диаграммы или спецификации процесса.

10) Проверка основных требований по DFD соответствующего уровня.

11) Добавление определений новых потоков в словарь данных при каждом их появлении на диаграммах.

12) Параллельное (с процессом декомпозиции) изучение требований (в том числе и вновь поступающих), разбиение их на элементарные и идентификация процессов или спецификаций процессов, соответствующих этим требованиям.

13) После построения двух-трех уровней проведение ревизии с целью проверки корректности и улучшения понимаемости модели.

14) Построение спецификации процесса (а не простейшей диаграммы) в случае, если некоторую функцию сложно или невозможно выразить комбинацией процессов.

7) Формирование DFD первого уровня на базе процессов предварительной контекстной диаграммы.

8) Проверка основных требований по DFD первого уровня.

9) Декомпозиция каждого процесса текущей DFD с помощью детализирующей диаграммы или спецификации процесса.

10) Проверка основных требований по DFD соответствующего уровня.

11) Добавление определений новых потоков в словарь данных при каждом их появлении на диаграммах.

12) Параллельное (с процессом декомпозиции) изучение требований (в том числе и вновь поступающих), разбиение их на элементарные и идентификация процессов или спецификаций процессов, соответствующих этим требованиям.

13) После построения двух-трех уровней проведение ревизии с целью проверки корректности и улучшения понимаемости модели.

14) Построение спецификации процесса (а не простейшей диаграммы) в случае, если некоторую функцию сложно или невозможно выразить комбинацией процессов.

7) Формирование DFD первого уровня на базе процессов предварительной контекстной диаграммы.

8) Проверка основных требований по DFD первого уровня.

9) Декомпозиция каждого процесса текущей DFD с помощью детализирующей диаграммы или спецификации процесса.

10) Проверка основных требований по DFD соответствующего уровня.

11) Добавление определений новых потоков в словарь данных при каждом их появлении на диаграммах.

12) Параллельное (с процессом декомпозиции) изучение требований (в том числе и вновь поступающих), разбиение их на элементарные и идентификация процессов или спецификаций процессов, соответствующих этим требованиям.

13) После построения двух-трех уровней проведение ревизии с целью проверки корректности и улучшения понимаемости модели.

14) Построение спецификации процесса (а не простейшей диаграммы) в случае, если некоторую функцию сложно или невозможно выразить комбинацией процессов.

IDEF3

Стандарт IDEF3 – это методология описания процессов, рассматривающих последовательность выполнения и причинно-следственные связи между ситуациями и событиями для структурного представления знаний о системе. Моделирование в стандарте IDEF3 производится с использованием графического представления процесса, материальных и информационных потоков в этом процессе, взаимоотношением между операциями и объектами в процессе. При помощи IDEF3 описывают логику выполнения работ, очередность их запуска и завершения, т.е. IDEF3 представляет инструмент моделирования сценариев действия сотрудников организации, отделов, цехов и т.п., например, порядок обработки заказа или события, на которые необходимо реагировать за конечное время, выполнение действия по производству товара и т.п.

Метод IDEF3 использует категорию сценариев для упрощения структуры описания сложного многоэтапного процесса. Под сценарием понимается повторяющаяся последовательность ситуаций или действий, которые описывают типичный класс проблем, присутствующих в организации или системе, описание последовательности изменения свойств объекта в рамках рассматриваемого процесса, например описание последовательности обработки менеджером заявки.

Метод позволяет проводить описание с необходимой степенью подробности посредством использования декомпозиции. IDEF3 как инструмент моделирования фиксирует следующую информацию о процессе:

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

Графические элементы. Это элементы, из которых состоят IDEF3-диаграммы описания процесса, включают функциональные элементы, связи старшинства, перекрестки, объекты ссылки и примечания.

Функциональные элементы. Описание процесса представляет всевозможные ситуации (процессы, функции, действия, акты, события, сценарии, процедуры, операции или решения), которые могут происходить в моделируемой системе в логических и временных отношениях. Каждый процесс представлен полем, отображающим название процесса (рис. 1). Номер идентификатора процесса назначается последовательно.

В правом нижнем углу функционального элемента располагается ссылка (IDEF0 или другие) и используется для указания ссылок либо на элементы из функциональной модели IDEF0, либо на указания на отделы или конкретных исполнителей, которые будут выполнять указанную работу.



Рис. 1. Синтаксис функционального элемента

Элемент связи. Элемент диаграммы описания процессов — связь — необходим для организации отношений между элементами диаграммы и описания динамики происходящих процессов. Связи используются прежде всего для обозначения отношений между функциональными элементами. Для отображения временной последовательности выполнения сценариев в диаграммах описания процесса используются два основных типа связи: связи старшинства и относительные связи (рис. 2). Связи между элементами в схеме процесса представляют отношения между соответствующими элементами в смоделированной системе. В схемах процесса можно использовать три типа стрелок: связь старшинства, объектный поток и относительная связь.

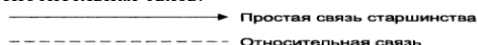


Рис. 2. Типы связей в IDEF3-диаграммах

Связь старшинства. Эта связь отражает временные отношения старшинства между элементами диаграммы. При этом первый элемент должен завершиться прежде, чем начнет выполняться следующий (рис. 3).

Относительная связь указывает на то, что между взаимодействующими элементами

диаграммы описания процесса существует отношение неопределенного типа (рис. 2).

Связь поток объектов выражает перенос одного или нескольких объектов от одного функционального элемента к другому и наследует все свойства простой связи старшинства (рис. 4).

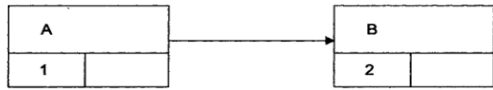


Рис. 3. Семантика использования связи старшинства

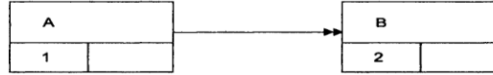
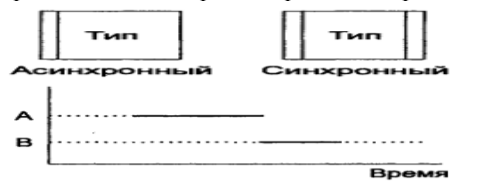


Рис. 4. Представление связи поток объектов

Перекресток. Перекрестки используются для отображения логики отношений между множеством событий и временной синхронизации активизации элементов IDEF3-диаграмм. Различают перекрестки для слияния и разветвления стрелок. Перекресток не может одновременно использоваться и для слияния и для разветвления. При внесении перекрестка в диаграмму необходимо указать его тип. Тип перекрестка определяет логику и временные параметры отношений между элементами диаграммы. Все перекрестки на диаграмме нумеруются, каждый номер имеет префикс «J». Тип перекрестка обозначается внутри элемента: & - логический И, О – логический ИЛИ, Х – логический перекресток неэквивалентности.

Стандарт IDEF3 предусматривает разделение перекрестков типа & и О на синхронные и асинхронные. Это разделение позволяет учитывать в диаграммах описания процессов синхронизацию времени активации (рис. 5).

График запуска – это визуальное отображение временной последовательности выполнения функциональных элементов. Возможный график для ситуации, представленной на рис. 3, приведен на рис. 6.



Методология IDEF3 использует 5 логических типов для моделирования возможных последовательностей действий процесса в сценарии (табл. 1).

Логические типы			
Обозначение	Наименование	Смысл в случае слияния стрелок (Fan-in Junction)	Смысл в случае разветвления стрелок (Fan-out Junction)
&	Asynchronous AND	Все предшествующие процессы должны быть завершены	Все следующие процессы должны быть запущены
&	Synchronous AND	Все предшествующие процессы завершены одновременно	Все следующие процессы запускаются одновременно
O	Asynchronous OR	Один или несколько предшествующих процессов должны быть завершены	Один или несколько следующих процессов должны быть запущены
O	Synchronous OR	Один или несколько предшествующих процессов завершаются одновременно	Один или несколько следующих процессов запускаются одновременно
X	XOR (Exclusive OR)	Только один предшествующий процесс завершен	Только один следующий процесс запускается

Правила создания перекрестков. На одной диаграмме IDEF3 может быть создано несколько перекрестков различных типов. Определенные сочетания перекрестков для

слияния и разветвления могут приводить к логическим несоответствиям.

Поэтому необходимо соблюдать следующие правила:

1. Каждому перекрестку для слияния должен предшествовать перекресток для разветвления.

2. Перекресток для слияния «И» не может следовать за перекрестком для разветвления типа синхронного или асинхронного «ИЛИ» (рис. 7).

Действительно, после работы 1 может запуститься только одна работа – 2 или 3, а для запуска работы 4 требуется окончание обеих работ – 2 и 3. Такой сценарий не может реализоваться.

3. Перекресток для слияния «И» не может следовать за перекрестком для разветвления типа исключающего «ИЛИ» (рис.8).

4. Перекресток для слияния типа исключающего «ИЛИ» не может следовать за перекрестком для разветвления типа «И» (рис.9).

5. Перекресток, имеющий одну стрелку на одной стороне, должен иметь более одной стрелки на другой.

На рис. 10 и 11 приведены примеры использования комбинаций перекрестков и возможные графики запуска для этих примеров.

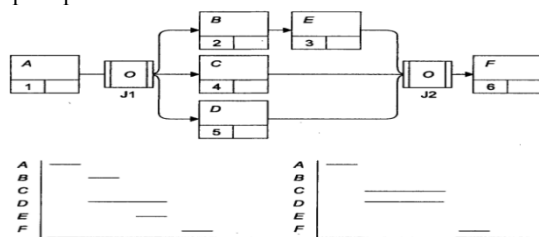


Рис. 10. Использование перекрестков синхронного OR и возможные графики запуска

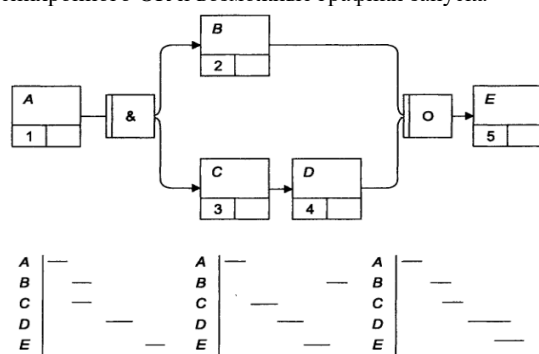


Рис. 11. Использование асинхронного AND-перекрестка разветвления и асинхронного OR-перекрестка слияния и возможные графики запуска

Элемент «референт». Элемент «референт» – это элемент ссылки. Референты расширяют границы понимания диаграммы и упрощают конструкцию описания, т.е. исключают неоднозначность.

Референты предназначены:

- для обращения к предварительно определенному функциональному элементу без дублирования его определения;
- передачи управления или определения возвратных циклов;
- организации связи между IDEF3-диаграммами.

Типы референтов приведены в таблице 2. Помимо деления на виды идеология IDEF3 определяет два вида референтов по способу запуска. На рис. 12 представлены их условные обозначения.

Типы референтов		
Тип референта	Обозначение референта	Locator
<i>UOB</i>	Имя функционального элемента <i>UOB</i>	№ <i>UOB</i>
<i>SCENARIO</i>	Название сценария	№ <i>Scenario</i>
<i>TS</i> (Transition Schematic)	Название диаграммы перехода состояний	№ Диаграммы перехода
<i>GO-TO</i>	Имя функционального элемента <i>UOB</i>	№ <i>UOB</i> , № сценария или декомпозиции, в которой находится элемент

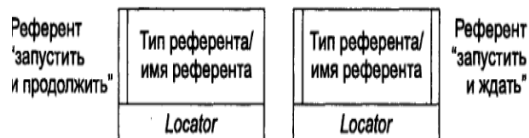


Рис. 9. Синтаксис референта

Методология IDEF3 дает возможность представлять процесс в виде иерархически организованной совокупности диаграмм. Диаграммы состоят из нескольких элементов описания процесса IDEF3, причем каждый функциональный элемент потенциально может быть детализирован на другой диаграмме. Такое разделение сложных комплексных процессов на их структурные части называется декомпозицией.

Декомпозиция формирует границы для описания процесса, и каждый функциональный элемент рассматривается как формальная граница некоторой части целой системы, которая описывает весь процесс. Декомпозированная диаграмма, называемая диаграммой-потомком, описывает процесс более детально. Декомпозируемый функциональный элемент называется родительским, а содержащая его диаграмма – соответственно родительская диаграмма. Понятие анализ означает разложение на составляющие. Но декомпозиция – это больше, чем разложение на части. Она включает также синтез. Подлинная декомпозиция заключается в начальном разделении объекта на более мелкие части и последующем соединении их в более детальное описание. Применяя принцип декомпозиции неоднократно, можно структурировать описание процесса до любого уровня подробности. Каждый функциональный элемент может иметь любое число различных декомпозиций на том же самом уровне детализации с целью представления различных точек зрения или обеспечения большей подробности при описании исходного процесса (рис. 13).

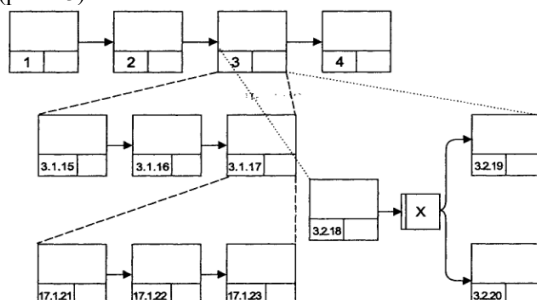


Рис. 13. Пример нумерации функциональных элементов при использовании декомпозиции и описания различных точек зрения на выполнение процессов