

ФУНКЦИОНАЛЬНОЕ МОДЕЛИРОВАНИЕ СИСТЕМ

МЕТОДИКИ:

SADT - IDEF0 , DFD, IDEF3

**Пакет прикладных программ
All Fusion Process Modeler**

1. Общие сведения о функционально-структурном подходе к моделированию бизнес-процессов

Функционально-структурный подход к моделированию бизнес-процессов предусматривает выделение в исследуемой системе путем ее декомпозиции процессы, подпроцессы, функции и определение их иерархии

Подход был предложен Дугласом Россом в 1960-х гг. в качестве методологии SADT (Structured Analysis and Design Technique), предусматривающий проведение декомпозиции анализируемого процесса и представление его как совокупности взаимосвязанных операций, каждая из которых имеет четко определенные входы и выходы, определяющие связи между операциями с учетом необходимых для их выполнения ресурсов

В 1970-х гг. методология SADT получила распространение, благодаря ее использованию Министерством обороны США в качестве поддержки производства ICAM (Integrated Computer-Aided Manufacturing).

Основной целью применения функционально-структурного подхода стало повышение эффективности производственного процесса за счет использования компьютерных технологий

**В последствии методология SADT
была переименована в IDEF
(ICAM DEFinition, далее –
Integrated DEFinition) и
утверждена в качестве
федерального стандарта США
«IDEF0». Последняя редакция
данного стандарта выпущена в
1993г.**

**На сегодняшний день стандарт IDEF0 получил
международное распространение и
используется для:**

- моделирования бизнес-процессов,**
- моделирования информационных систем,**
- разработки моделей приложений,**
- разработки различных нормативно-правовых документов (положений, регламентов, инструкций и др.).**

В семействе IDEF выделяют следующие методологии:

- IDEF0** – используется для функционального моделирования бизнес-процессов верхнего уровня, образующих общую структуру процесса;
- IDEF1** – методология моделирования внутрисистемных информационных потоков;
- IDEF1X** – методология моделирования реляционных структур;
- IDEF3** – методология моделирования атомарных функций (не подлежащих дальнейшей декомпозиции) в виде потоков работ;
- IDEF4** – методология объектно-ориентированного моделирования систем в виде классов, диаграмм наследования и др.
- IDEF5** – онтологическое моделирование сложных систем путем создания словаря терминов и правил для формулирования и описания системы в определенный момент времени

Для описания потоков данных между компонентами исследуемой системы (процессами, подпроцессами) используется методология DFD (Data Flow Diagrams) – диаграммы потоков данных. Данная методология используется для детализации процессов, представленных в IDEF0 и IDEF3

1. Методика IDEF0

Методология функционально-структурного моделирования IDEF0 основана на построении структуры функций, которые выполняются системой с определением входов, выходов, механизмов и управления для каждой функции

Функциональная модель в нотации IDEF0

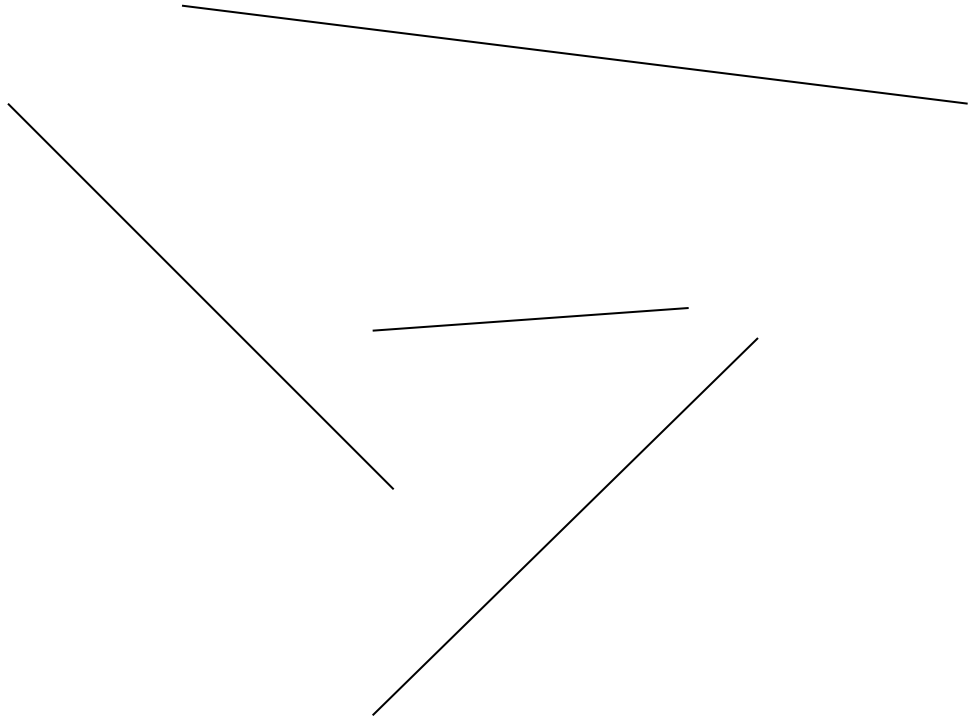
Нотацию IDEF0 определяют следующие правила:

- 1. Функция изображается в виде прямоугольника (Activity), в правом нижнем углу которого приведен ее номер.**
- 2. Левая сторона блока Activity используется для изображения входов функции в виде стрелок.**
- 3. Из правой стороны блока Activity в виде стрелок изображаются выходы системы.**
- 4. В нижнюю сторону блока Activity входят стрелки, изображающие механизмы функции.**
- 5. В верхней части функции определяются способы управления функцией.**

- 6. Каждой функции присваивается имя. Имя функции всегда должно содержать глагольную форму, подразумевающее действие, направленное на преобразование входов функции в ее выходы и совершаемое указанными механизмами.**
- 7. Каждая функция должна иметь минимум по одному входу, выходу, механизму и управлению.**
- 8. Каждой стрелке присваивается имя в форме существительного. Имя должно иметь лаконичное название, конкретно характеризующее назначение входа, выхода, механизма или управления.**

- 9. Стрелки, изображающие входы, выходы, механизмы и управление, называются «граничными». Такое название обуславливается тем, что каждая из них идет от границы модели. В качестве границы модели выступает внешняя среда, а для дочерних диаграмм – границы родительской.**
- 10. Модель, построенная в нотации IDEF0, имеет иерархическую древовидную структуру, каждый узел которой представляет собой диаграмму. Вершина древовидной структуры называется ТОР-диаграммой, каждый последующий узел – диаграммами декомпозиции.**

Принцип декомпозиции в нотации IDEF0



Каждая диаграмма в нотации IDEF0 имеет свой уникальный номер, представленный в левой нижней части, а в правой верхней части - контекст декомпозиции. Так, например, TOP-диаграмма имеет номер А-0. Буква «А» в индексе образована от Activity, «0» - уровень декомпозиции, контекст – «TOP-диаграмма»

Номер диаграммы декомпозиции «А-0» обозначается индексом «А0» (без дефиса), имя диаграммы образуется по названию Activity на ТОР-диаграмме (в приведенном примере – «Функция»), а контекст представлен блоком Activity с черной заливкой

Номера диаграмм последующих уровней декомпозиции формируются по принципу «А1», «А2», «А3» и т.д. Так, например, на приведенном рисунке нижней диаграмме присвоен индекс «А3» ввиду того, что она изображает результат декомпозиции блока Activity с номером «3», что и изображено в поле «Context». Имя диаграммы «А3» образовано по имени соответствующего блока Activity на родительской диаграмме

При дальнейшей декомпозиции блоков Activity будут образовываться номера «A31», «A32», «A314» и т.д. Так, например, номер диаграммы «A314» говорит о том, что на ней приведен результат декомпозиции четвертого блока Activity диаграммы «A31», а сама диаграмма представляет пятый уровень декомпозиции модели

В процессе декомпозиции на новую диаграмму по умолчанию добавляется четыре блока Activity, что связано с оптимальным количеством функциональных блоков, которые может содержать диаграмма. И хотя количество блоков Activity, содержащихся на диаграмме, не ограничено, их большое количество может привести к непониманию и неудобочитаемости модели

Нотация IDEF0 предусматривает четыре вида граничных стрелок:

1. Вход. В качестве входов функциональных блоков Activity могут выступать различного рода информация, документы, материальные объекты, которые будут трансформированы в результате выполнения функции. Например, документ, который будет подписан, информация, которая будет обработана, сырье, которое будет переработано в полуфабрикат или готовый продукт.

2. Выход. В качестве выходов функциональных блоков Activity выступают трансформированные или измененные в результате выполнения функции документы, информация или материальные объекты. Например, созданный документ, подписанный документ, обработанная информация, полуфабрикат или готовая продукция, товар или услуга

3. Механизмы. В качестве механизмов функции выступают различные ресурсы, с помощью которых она может быть выполнена (человек, оборудование, инструмент, приложение, финансы, энергетические ресурсы и др.). **Например,** если в качестве входа функции определен документ, а в качестве выхода – подписанный документ, тогда механизмом этой функции будет человек, который подписывает документ.

4. Управление. В качестве управления для функции чаще всего выступают документы, регламентирующие ее выполнение. Например, положение, инструкция, рецептура, методические указания, локальные нормативно-правовые акты (приказы, распоряжения, решения и др.), федеральные, краевые нормативно-правовые акты и др.

Граничные стрелки в методологии IDEF0 можно представлять укрупненно, а также при необходимости проводить их детализацию. В первую очередь это предусматривается для обеспечения удобочитаемости диаграммы

Так, например, на ТОР-диаграмме не имеет смысла приводить перечень всех должностей персонала организации, а достаточно привести одну стрелку «Персонал», а потом на диаграммах декомпозиции уточнить соответствующую должность.

Аналогичный прием, можно провести с входами, выходами и управлением

Детализация граничных стрелок модели

В процессе корректировки и уточнения модели исследователь может добавлять или удалять граничные стрелки на диаграммах, что приводит к их обрыву. Граничная стрелка с обрывом изображается в квадратных скобках []

Обрыв граничных стрелок модели

Квадратные скобки на границе диаграммы обозначают, что граничная стрелка оборвана на родительской диаграмме. Если квадратные скобки изображены возле блока Activity – это говорит о том, что граничная стрелка оборвана на дочерней диаграмме

Обрыв граничной стрелки – это синтаксическая ошибка модели. Намеренный обрыв или невыведение граничной стрелки на родительскую диаграмму должен сопровождаться «туннелированием». Затуннелированная граничная стрелка изображается в круглых скобках.

Намеренный обрыв граничной стрелки чаще всего производится при смене нотаций моделирования

Туннелирование граничных стрелок модели

**Рассмотрим процесс создания модели
бизнес-процесса с использованием
методологии IDEF0 на примере бизнес-
процессов**

**ЗАО «Мясоперерабатывающего
комплекса «Динской», который
представляет собой многопрофильную
группу сельскохозяйственных,
производственных, торгово-сбытовых
компаний для совместной
хозяйственной деятельности**

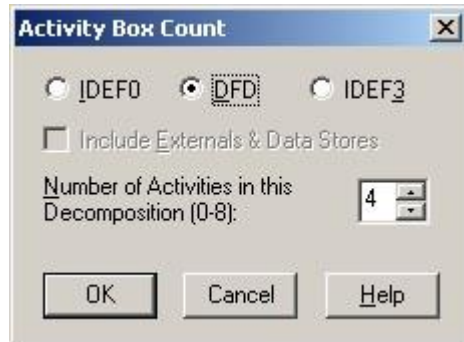
Методика DFD

Диаграммы потоков данных (Data Flow diagramming, DFD) используются для описания документооборота и обработки информации. Подобно IDEF0, DFD представляет модельную систему как есть связанных между собой функциональных блоков. Их можно использовать как дополнение к модели IDEF0 для более наглядного отображения текущих операций документооборота в корпоративных системах обработки информации.

DFD описывает:

- функции обработки информации;**
- документы (стрелки), объекты, сотрудников или отделы, которые участвуют в обработке информации;**
- внешние сущности (External references), которые обеспечивают интерфейс с внешними объектами, находящимися за границами моделируемой системы;**
- таблицы для хранения документов (хранилище данных, data store).**

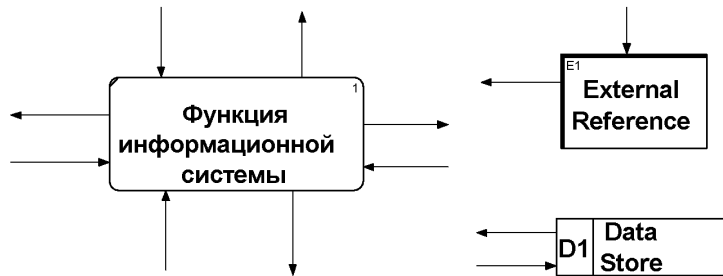
Для того чтобы дополнить функциональную модель диаграммой DFD, необходимо выделить функцию, которую нужно декомпозировать и на панели инструментов нажать кнопку Go to Child Diagram при этом ▼ откроется окно Activity Box Count, в котором необходимо указать методологию – DFD и выбрать количество функциональных блоков.



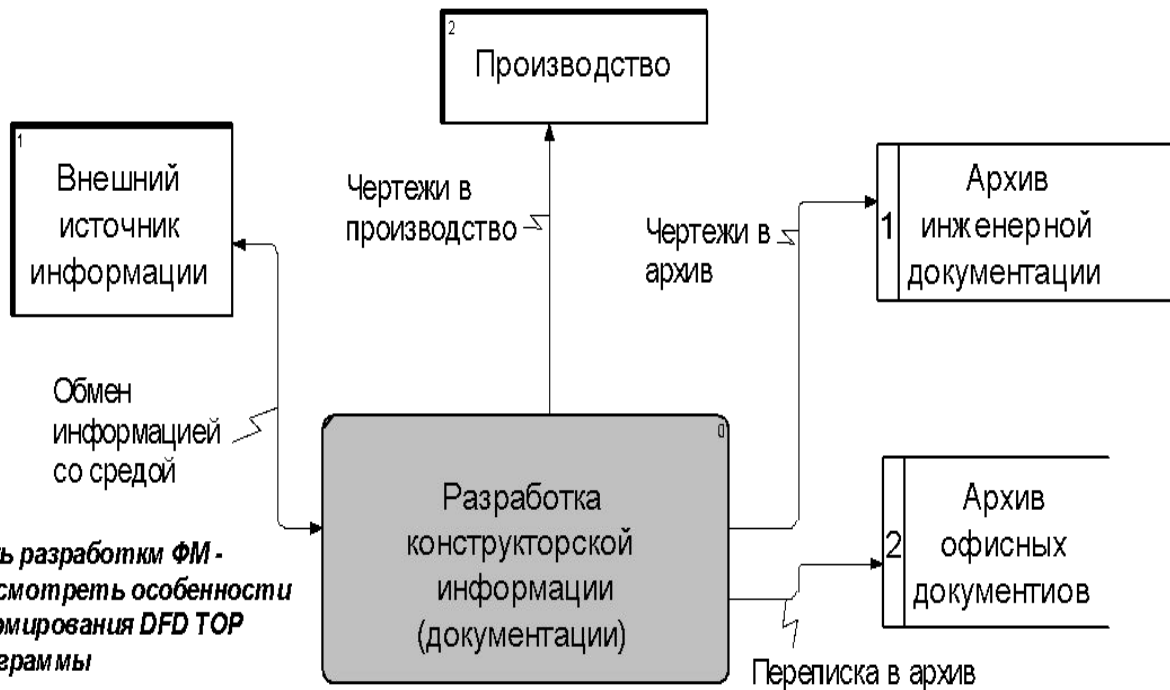
DFD. Операционные элементы.

Графические средства формирования функциональной модели

DFD.
Структурное
моделирование
информационных
систем



TOP диаграмма DFD



Цель разработки ФМ - рассмотреть особенности формирования DFD TOP диаграммы

Функциональные блоки

В DFD методологии, функциональные блоки представляют собой функции системы, преобразующие входы в выходы.

Функциональные блоки изображаются в виде прямоугольников со скругленными углами, имеют входы и выходы, но не поддерживают механизмы и управления.

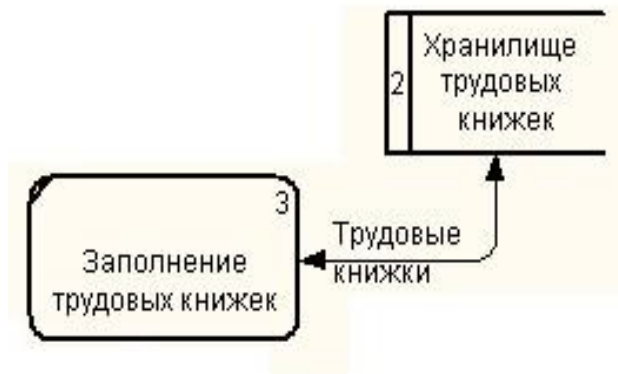
Для добавления функциональных блоков на диаграмму используется кнопка Activity Box Tool



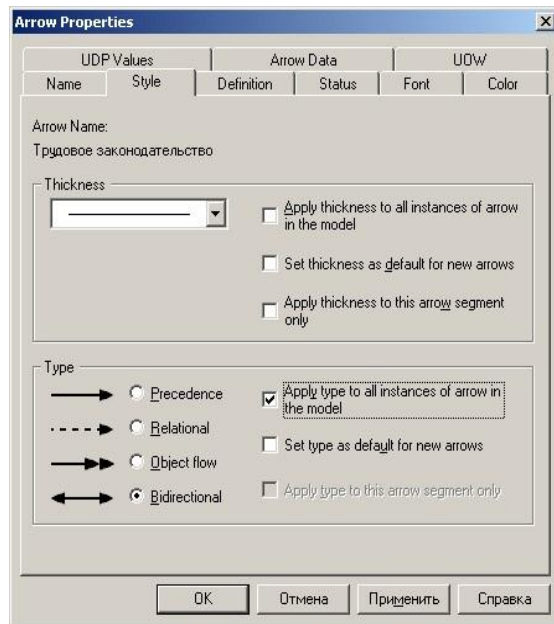
Стрелки

Описывают движение объектов из одной части системы в другую. Поскольку в DFD каждая сторона Activity не имеет четкого назначения, как в IDEF0, стрелки могут подходить и выходить из любой грани прямоугольника функционального блока. В DFD также применяются двунаправленные стрелки для описания диалогов типа команды – ответа между функциональными блоками, между функциональными блоками и внешними сущностями

Двунаправленная стрелка между функциональным блоком и хранилищем данных

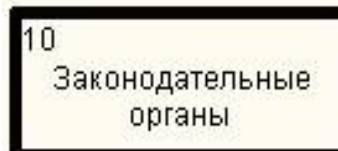



**Двунаправленная
стрелка создается с
помощью вызова
диалогового окна
свойств уже созданной
стрелки, на вкладке
Style которого
указывается тип
стрелки Bidirectional**



Внешние сущности

Изображают входы в систему и/или выходы из системы. Внешние сущности изображаются в виде прямоугольника с тенью и обычно располагаются по краям диаграммы

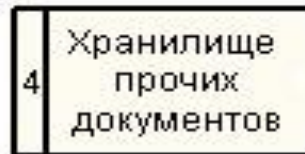


Одна внешняя сущность может быть использована многократно на одной или нескольких диаграммах. Обычно такой прием применяют, чтобы не рисовать слишком длинных и запутанных стрелок. Для добавления внешней сущности используется External Reference Tool 

Внешняя сущность является источником или потребителем данных извне модели

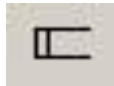
Хранилища данных

В отличие от стрелок, описывающих объекты в движении, хранилища данных изображают объекты в покое. В материальных системах хранилища данных изображаются там, где объекты ожидают обработки, например в очереди. В системах обработки информации хранилища данных являются механизмом, которые позволяют сохранить данные для последующих процессов



**Одноименные хранилища данных
также могут быть использованы
многokrратно на одной или нескольких
диаграммах.**

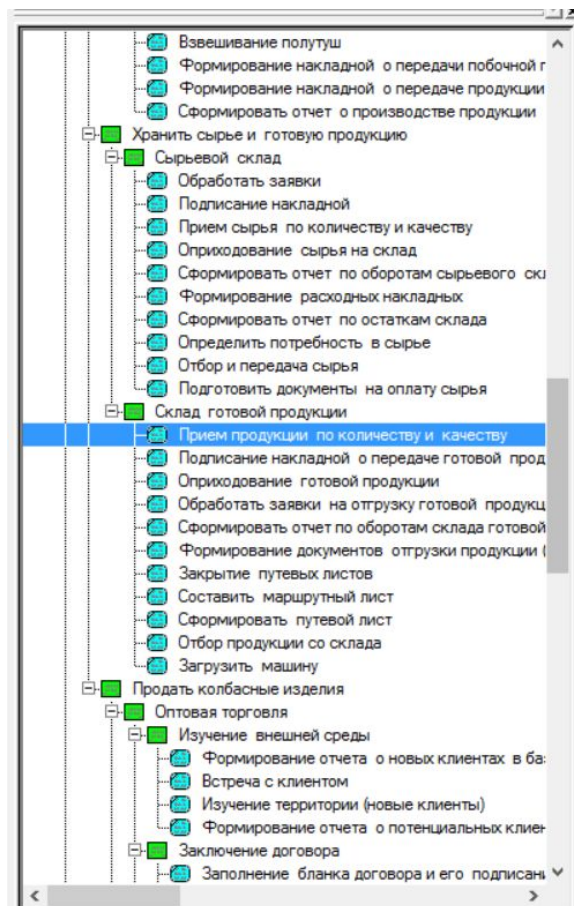
**Для добавления хранилища данных,
используется Data Store Tool**

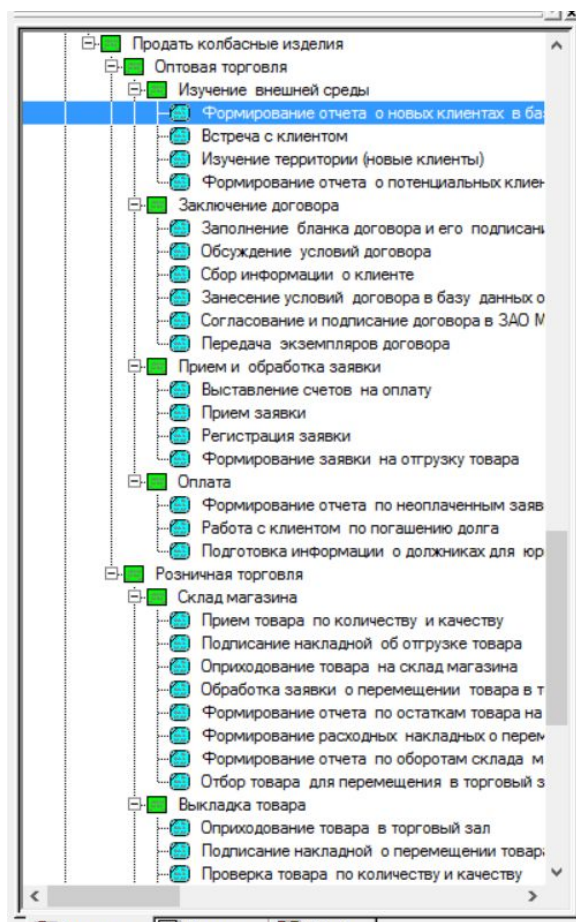


Хранилища данных могут иметь как стандартный вид отображения, так и настраиваемый.



Для того чтобы изменить вид, необходимо открыть свойства объекта и на вкладке Box Style переключить переключатель в положение Custom и из выпадающего списка выбрать интересующее изображение. Чтобы на объекте было видно его название, необходимо поставить галочку в пункте Show Name





Методика IDEF3

Для описания логики взаимодействия информационных потоков более подходит IDEF3, называемая также workflow diagramming – методологией моделирования, использующая графическое описание информационных потоков, взаимоотношений между процессами обработки информации и объектов, являющихся частью этих процессов

Диаграммы Workflow могут быть использованы в моделировании бизнес – процессов для анализа завершенности процедур обработки информации. С их помощью можно описывать сценарии действий сотрудников организации, например последовательность обработки заказа или события, которые необходимо обработать за конечное время. Каждый сценарий сопровождается описанием процесса и может быть использован для документирования каждой функции.

IDEF3 – это метод, имеющий
основной целью дать
возможность аналитикам
описать ситуацию, когда
процессы выполняются в
определенной
последовательности, а также
описать объекты, участвующие
совместно в одном процессе

Техника описания набора данных IDEF3 является частью структурного анализа. В отличие от некоторых методик описаний процессов IDEF3 не ограничивает аналитика чрезмерно жесткими рамками синтаксиса, что может привести к созданию неполных или противоречивых моделей

IDEF3 может быть также использован как метод создания процессов. IDEF3 дополняет IDEF0 и содержит все необходимое для построения моделей, которые в дальнейшем могут быть использованы для имитационного анализа

Каждый функциональный блок в IDEF3 описывает какой-либо сценарий бизнес-процесса и может являться составляющей другого функционального блока. Поскольку сценарий описывает цель и рамки модели, важно, чтобы функциональные блоки именовались отглагольным существительным, обозначающим процесс действия.

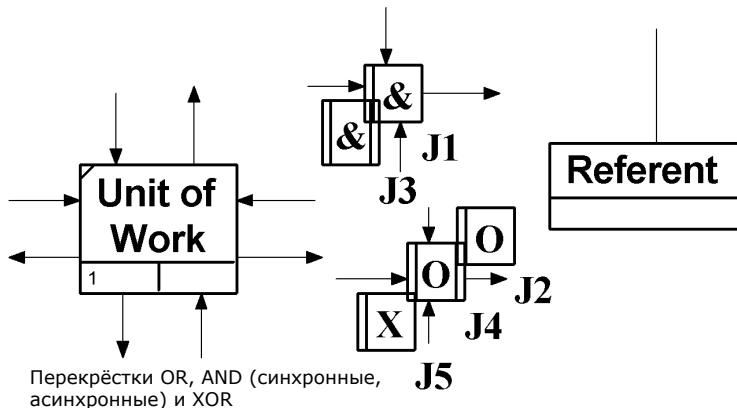
IDEF3. Операционные элементы.

Графические средства формирования функциональной модели

IDEF3.

Потоковое моделирование материальных и информационных систем.

Последовательная декомпозиция от TOP диаграммы без миграции стрелок. Ограничений на топологическую структуру нет. Слияния – разветвления стрелок – только через перекрёстки.



Единицы работы - Unit of Work (UOW).

UOW, также называемые работами (действиями), являются центральными компонентами модели. В

IDEF3 функциональные блоки изображаются прямоугольниками с прямыми углами и имеют имя, выраженным отглагольным существительным, обозначающим процесс действия, одиночным или в составе словосочетания, и номер (идентификатор); другое имя существительное в составе того же словосочетания, зависимое от отглагольного существительного, обычно отображает основной выход (результат) функционального блока (например: «Изготовление изделия»).

Часто имя существительное в имени функционального блока меняется в процессе моделирования, поскольку модель может уточняться и редактироваться. Идентификатор функционального блока присваивается при создании и не меняется никогда. Даже если функциональный блок будет удален, его идентификатор не будет вновь использоваться для других блоков

СВЯЗИ

Связи показывают взаимоотношения функциональных блоков. Все связи в IDEF3 однонаправлены и могут быть направлены куда угодно, но обычно диаграммы IDEF3 стараются построить так, чтобы связи были направлены слева направо

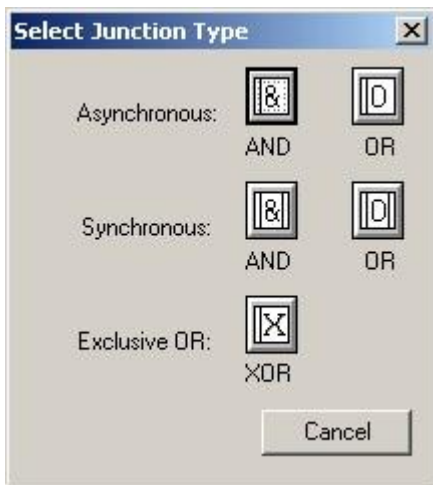
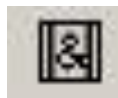
Перекрестки (Junction)

Окончание одного действия может служить сигналом к началу нескольких действий, или же одно действие для своего запуска может ожидать окончания нескольких действий.

Перекрестки используются для отображения логики взаимодействия стрелок при слиянии и разветвлении или для отображения множества событий, которые могут или должны быть завершены перед началом следующего действия. Различают перекрестки для слияния (Fan-in Junction) и разветвления (Fan-out junction) стрелок.

Перекресток не может использоваться одновременно для слияния и разветвления.

Для внесения перекрестка в диаграмму, служит кнопка Junction Tool на панели инструментов. В диалоговом окне Select Junction Style, необходимо указать тип перекрестка.



Обозначение	Наименование	Смысл в случае слияния стрелок (Fan-in Junction)	Смысл в случае разветвления стрелок (Fan-out Junction)
	Asynchronous AND (Асинхронное И)	Все предшествующие процессы должны быть завершены	Все следующие процессы должны быть запущены
	Synchronous AND (Синхронное И)	Все предшествующие процессы завершены одновременно	Все следующие процессы запускаются одновременно
	Asynchronous OR (Асинхронное ИЛИ)	Один или несколько предшествующих процессов должны быть завершены	Один или несколько следующих процессов должны быть запущены

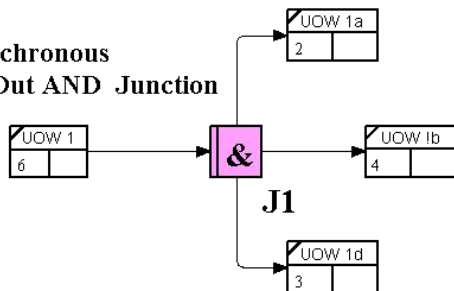
Обозначение	Наименование	Смысл в случае слияния стрелок (Fan-in Junction)	Смысл в случае разветвления стрелок (Fan-out Junction)
	Synchronous OR (Синхронное ИЛИ)	Один или несколько предшествующих процессов завершаются одновременно	Один или несколько следующих процессов запускаются одновременно
	XOR (Exclusive OR) (Исключающее ИЛИ)	Только один предшествующий процесс завершен	Только один следующий процесс запускается

Все перекрестки на диаграмме по умолчанию нумеруются, каждый номер имеет префикс J

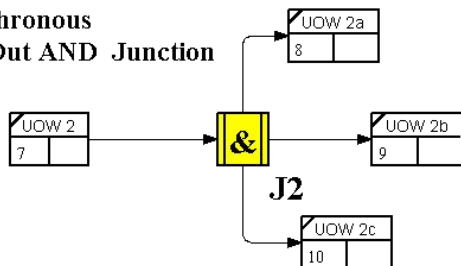
Типы AND перекрёстков

fan-out - A junction that depicts the split or divergence of a process into multiple alternative processing paths.

Asynchronous Fan Out AND Junction

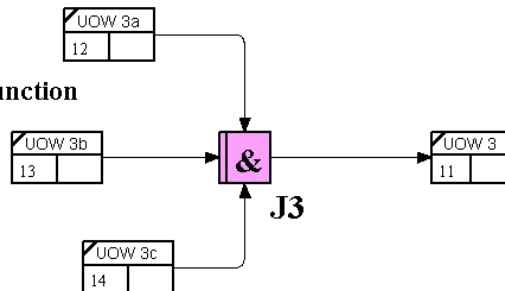


Synchronous Fan Out AND Junction

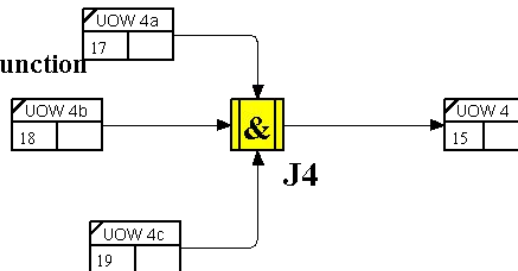


fan-in - A junction that depicts the joining or convergence of multiple processing paths into a single process.

Asynchronous Fan In AND Junction

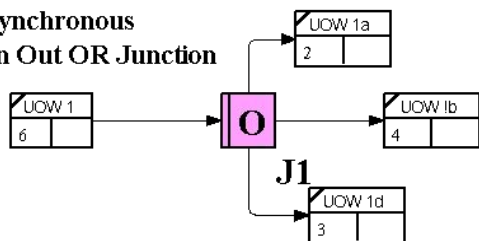


Synchronous Fan In AND Junction

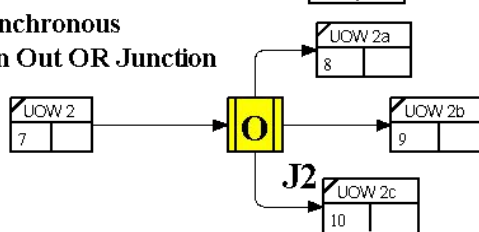


Типы OR перекрёстков

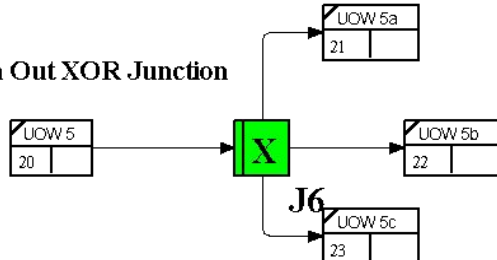
Asynchronous
Fan Out OR Junction



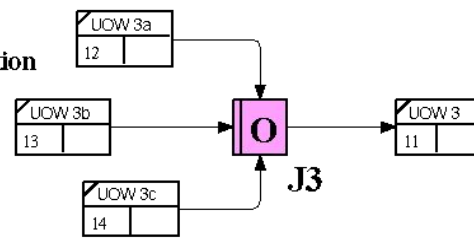
Synchronous
Fan Out OR Junction



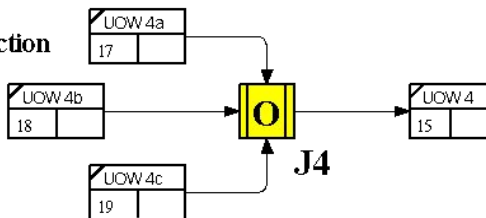
Fan Out XOR Junction



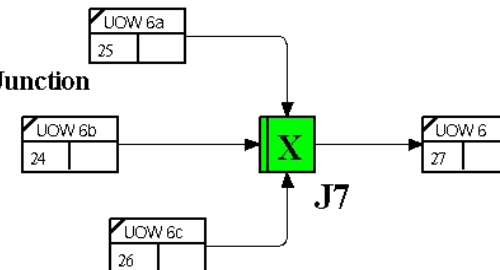
Asynchronous
Fan In OR Junction

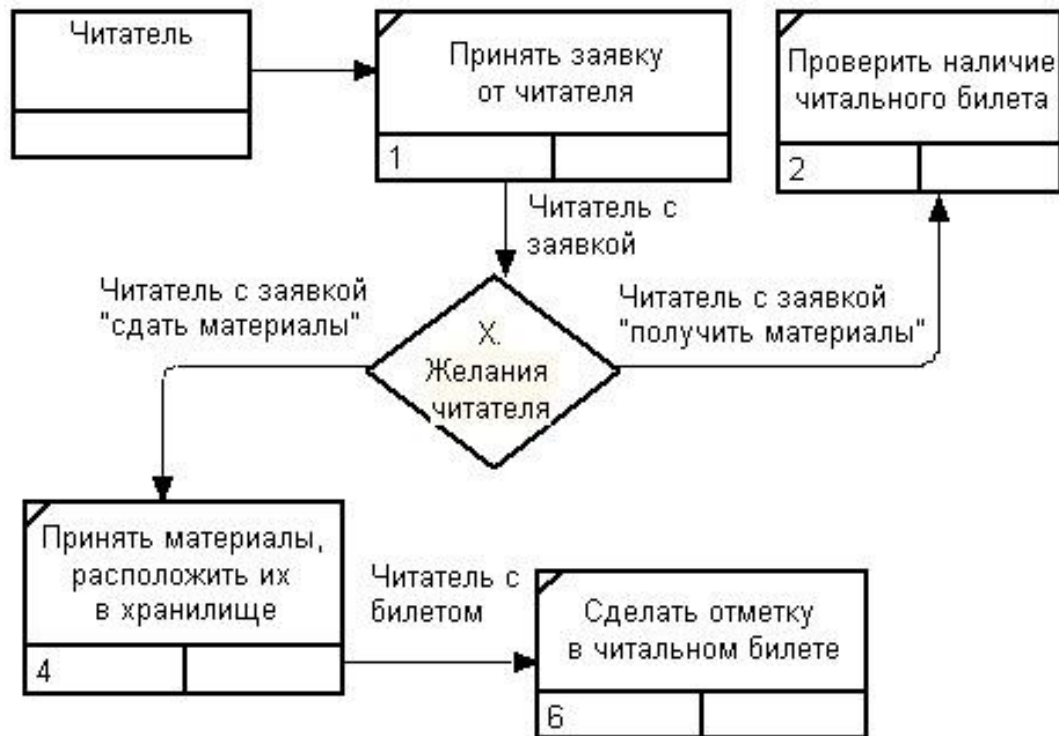


Synchronous
Fan In OR Junction



Fan In XOR Junction





Правила создания перекрестков

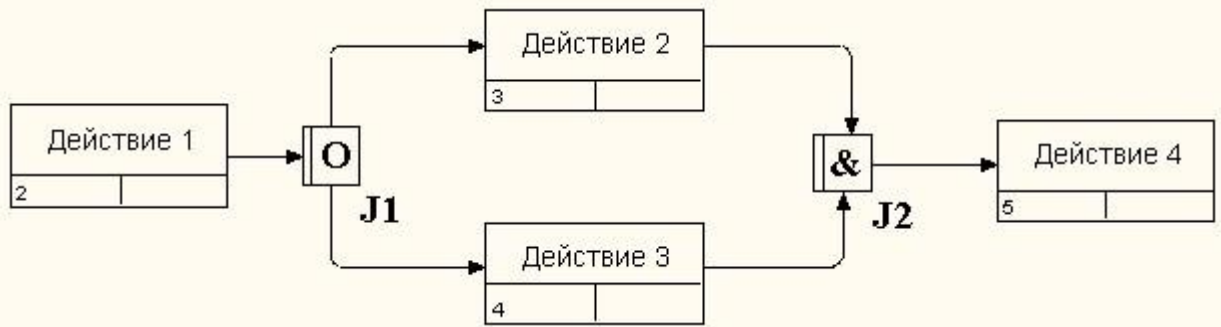
На одной диаграмме IDEF3 может быть создано несколько перекрестков различных типов. Определенные сочетания перекрестков для слияния и разветвления могут приводить к логическим несоответствиям.

**Чтобы избежать конфликтов, необходимо
соблюдать следующие правила:**

Каждому перекрестку для слияния должен предшествовать перекресток для разветвления.

Перекресток для слияния «И» не может следовать за перекрестком для разветвления типа синхронного или асинхронного «ИЛИ».

Пример неверного размещения перекрестков



Действительно, после действия 1 может запускаться только одно действие – 2 или 3, а для запуска действия 4 требуется окончание обоих действий – 2 и 3. Такой сценарий не может реализоваться

Пример неверного размещения перекрестков



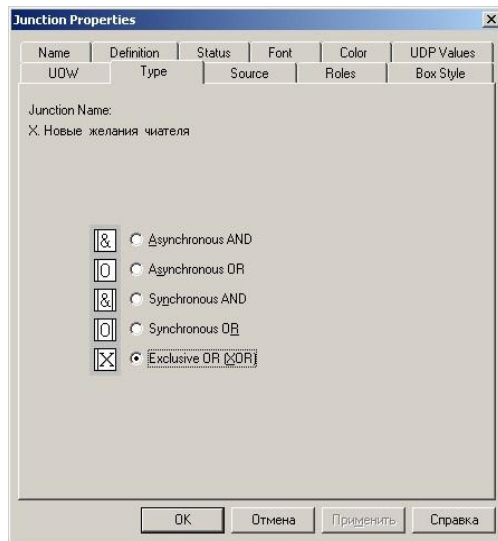
Перекресток для слияния «И» не может следовать за перекрестком для разветвления типа исключающего «ИЛИ»

Пример неверного размещения перекрестков



Перекресток для слияния типа исключающего «ИЛИ» не может следовать за перекрестком для разветвления типа «И». В данном случае после завершения действия 1 запускаются оба действия – 2 и 3, а для запуска действия 4 требуется, чтобы завершилась одно и только одно действие – 2 или 3.

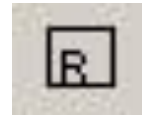
**Тип перекрестка
можно всегда
изменить в процессе
работы. Во вкладке
Type диалогового
окна Junction
Properties,
необходимо выбрать
интересующий тип и
нажать кнопку ОК.**



Объект ссылки

Объект ссылки в IDEF3 выражает некую идею, концепцию или данные, которые нельзя связать со стрелкой, перекрестком или действием. Для внесения объекта ссылки в диаграмму, служит кнопка Referent

Tool



**Объект ссылки
изображается в виде
прямоугольника**



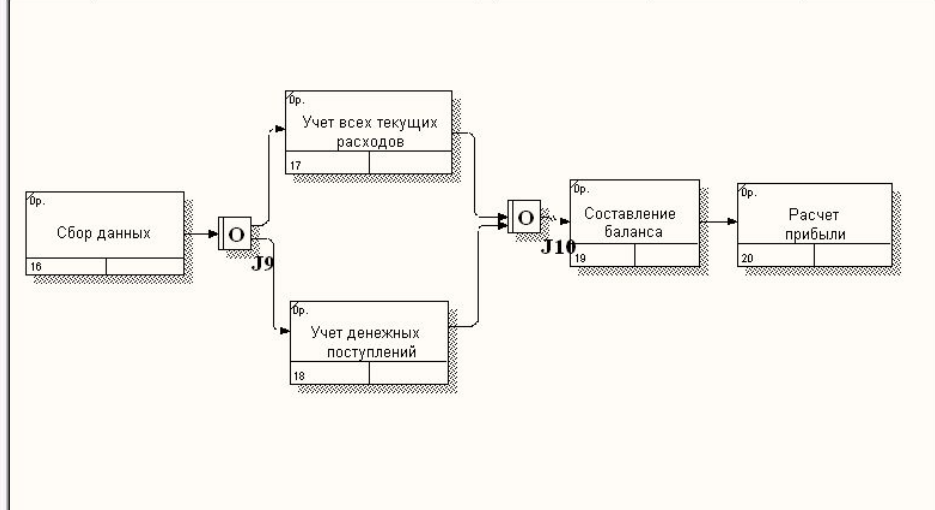


File Edit View Diagram Dictionary Model ModelMart Tools Window Help

70%

- Разработка маркет
- Организация работы с
- Запись пациента
 - Проверка и внес
 - Запись на прие
- Прием пациента
 - Проверка данны
 - Оформление нов
 - Просмотр предь
 - Осмотр
 - Назначение лече
 - Продолжение ле
 - Лечение
 - Записать резул
- Учет и оценка деятель
- Оплата
 - Сделать расчет
 - Произвести расч
 - Получение нали
 - Подсчет наличн
 - Распечатать чек
 - Внести данные в
- Учет лекарственных
- Просмотр о нали
- Поиск необходи
- Заказ поставщи
- Оформление за
- Финансовая отчетн
 - Сбор данных
 - Учет всех текущи
 - Учет денежных
 - Составление ба
 - Расчет прибыли

USED AT:	AUTHOR: Гурдина Людмила Валерьевна	DATE: 27.01.2009	WORKING	READER	DATE	CONTEXT:
	PROJECT: Модель деятельности стоматологического кабинета	REV: 27.01.2009	DRAFT			
	NOTES: 1 2 3 4 5 6 7 8 9 10		RECOMMENDED			
			PUBLICATION			A3



NODE:	TITLE:	NUMBER:
A33.1	ФИНАНСОВАЯ ОТЧЕТНОСТЬ	

