МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ

РОССИЙСКОЙ ФЕДЕРАЦИИ

МИРЭА - РОССИЙСКИЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ

**Кошкин Д.Е., Мороз Ю.В., Шемончук Д.С.**

**Моделирование бизнес-процессов**

**практикум**

**ДЛЯ СТУДЕНТОВ, ОБУЧАЮЩИХСЯ ПО НАПРАВЛЕНИЯМ ПОДГОТОВКИ 38.03.04 и 38.03.05**

Москва – 2018

УДК 004.62

ББК 32.97 М74

**Мороз Ю.В.** **Моделирование бизнес-процессов** [Электронный ресурс]: Практикум для студентов, обучающихся по направлениям подготовки 38.03.04 и 38.03.05 (первая часть) / Кошкин Д.Е., Мороз Ю.В., Шемончук Д.С. — М.: МИРЭА — Российский технологический университет, 2018. — 1 электрон. опт. диск (CD-ROM).

Практикум (лабораторный) разбит на две части. В настоящем издании представлена первая часть практикума.

Первая часть содержит справочную информацию по вопросам моделирования в среде CA ERwin Process Modeler, а также рекомендации и задания для выполнения лабораторных работ по построению моделей предметных областей в методологиях IDEF0, IDEF3 и DFD.

Практикум по выполнению лабораторных работ по дисциплине «Моделирование бизнес-процессов» предназначен для студентов, обучающихся по направлениям подготовки 38.03.04 «Государственное и муниципальное управление» и 38.03.05

«Бизнес-информатика», а также для студентов других направлений подготовки и специальностей, желающих освоить методологии IDEF0, IDEF3 и DFD или CASE- средство CA ERwin Process Modeler.

Практикум издается в авторской редакции.

Авторский коллектив: Кошкин Дмитрий Евгеньевич, Мороз Юлия Владимировна, Шемончук Дмитрий Сергеевич.

Рецензент:

Гавриленко Т. Ю. – К.э.н, заведующий кафедрой бизнес-технологий и управления МИРЭА — Российский технологический университет

Минимальные системные требования:

Наличие операционной системы Windows, поддерживаемой производителем. Наличие свободного места в оперативной памяти не менее 128 Мб.

Наличие свободного места в памяти хранения (на жестком диске) не менее 30 Мб. Наличие интерфейса ввода информации.

Дополнительные программные средства: программа для чтения pdf-файлов (Adobe Reader). Подписано к использованию по решению Редакционно-издательского совета

МИРЭА — Российский технологический университет от 12 ноября 2018 г. Обьем: 2.2 мб

Тираж 10

Нелегальное копирование и использование данного продукта запрещено. Электронное издание, номер государственной регистрации 0321900485 от 06.03.2019г.

© Кошкин Д.Е., Мороз Ю.В., Шемончук Д.С., 2018

**©** МИРЭА — Российский технологический университет, 2018

# Оглавление

[ВВЕДЕНИЕ 4](#_bookmark0)

1. [CA ERWIN PROCESS MODELER 6](#_bookmark1)
	1. [Описание CA ERwin Process Modeler 6](#_bookmark2)
	2. [Назначение CA ERwin Process Modeler 7](#_bookmark3)
	3. [Описание интерфейса CA ERwin Process Modeler 7](#_bookmark4)
2. [МЕТОДОЛОГИЯ IDEF0 12](#_bookmark5)
	1. [Работа c функциональными блоками 12](#_bookmark6)
	2. [Работа со стрелками 15](#_bookmark7)
	3. [Проверка синтаксиса модели 26](#_bookmark8)
	4. [Формирование отчета Node Tree 26](#_bookmark9)
	5. [Лабораторная работа № 1. Создание контекстной диаграммы и](#_bookmark10)

[диаграмм декомпозиции 28](#_bookmark10)

* 1. [Лабораторная работа № 2. Создание диаграммы «Дерево узлов» и](#_bookmark11) [диаграммы «Только для экспозиции» 39](#_bookmark11)
	2. [Лабораторная работа № 3. Функционально-стоимостное](#_bookmark12)

[моделирование 46](#_bookmark12)

* 1. [Лабораторная работа № 4. Анализ пользовательских свойств 52](#_bookmark13)
1. [МЕТОДОЛОГИЯ IDEF3 58](#_bookmark14)
	1. [Основные элементы методологии IDEF3 58](#_bookmark15)
	2. [Лабораторная работа № 5. Функциональное моделирование](#_bookmark16)

[в методологии IDEF3 63](#_bookmark16)

* 1. [Лабораторная работа № 6. Разработка диаграммы Swim Lane на](#_bookmark17)

[основе ролей производственного подразделения 66](#_bookmark17)

1. [МЕТОДОЛОГИЯ DFD 72](#_bookmark18)
	1. [Основные элементы методологии DFD 72](#_bookmark19)
	2. [Лабораторная работа № 7. Построение диаграммы декомпозиции в](#_bookmark20) [нотации DFD 76](#_bookmark20)
2. [ЗАДАНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ 81](#_bookmark21)

[БИБЛИОГРАФИЧЕСКИЙ СПИСОК 82](#_bookmark22)

# ВВЕДЕНИЕ

Создание современных информационных систем представляет собой сложнейшую задачу, решение которой требует применения специальных методик и инструментов. Неудивительно, что в последнее время среди системных аналитиков и разработчиков значительно вырос интерес к CASE- технологиям и инструментальным CASE-средствам, позволяющим максимально систематизировать и автоматизировать все этапы разработки программного обеспечения.

Технология создания информационных систем предъявляет особые требования к методикам реализации и программным инструментальным средствам, а именно: реализацию проектов по созданию информационных систем принято разбивать на стадии анализа (прежде чем создавать информационную систему, необходимо понять и описать бизнес-логику предметной области), проектирования (необходимо определить модули и архитектуру будущей системы), непосредственно кодирования, тестирования и сопровождения. Известно, что исправление ошибок, допущенных на предыдущей стадии, обходится примерно в 10 раз дороже, чем на текущей, откуда следует, что наиболее критическими являются первые стадии проекта. Поэтому крайне важно иметь эффективные средства автоматизации ранних этапов реализации проекта.

Проект по созданию сложной информационной системы невозможно организовать в одиночку. Коллективная работа существенно отличается от индивидуальной, поэтому при реализации крупных проектов необходимо меть средства координации и управления коллективом разработчиков.

Жизненный цикл создания сложной информационной системы сопоставим с ожидаемым временем ее эксплуатации. Другими словами, в современных условиях компании перестраивают свои бизнес-процессы примерно раз в два года, столько же требуется (если работать в традиционной технологии) для создания информационной системы. Может оказаться, что к

моменту сдачи информационной системы она уже никому не нужна, поскольку компания, ее заказавшая, вынуждена перейти на новую технологию работы. Следовательно, для создания информационной системы жизненно необходим инструмент, значительно (в несколько раз) уменьшающий время разработки информационной системы.

Вследствие значительного жизненного цикла может оказаться, что в процессе создания системы внешние условия изменились. Обычно внесение изменений в проект на поздних этапах создания информационной системы – весьма трудоемкий и дорогостоящий процесс. Поэтому для успешной реализации крупного проекта необходимо, чтобы инструментальные средства, на которых он реализуется, были достаточно гибкими к изменяющимся требованиям.

На современном рынке средств разработки информационных систем достаточно много систем, в той или иной степени удовлетворяющих перечисленным требованиям.

# 1 CA ERWIN PROCESS MODELER

* 1. Описание CA ERwin Process Modeler

CA ERwin Process Modeler (ранее BPwin) – инструмент для моделирования, анализа, документирования и оптимизации бизнес-процессов. CA ERwin Process Modeler можно использовать для графического представления бизнес-процессов. Графически представленная схема выполнения работ, обмена информацией, документооборота визуализирует модель бизнес- процесса. Графическое изложение этой информации позволяет перевести задачи управления организацией из области сложного ремесла в сферу инженерных технологий.

CA ERwin Process Modeler помогает четко документировать важные аспекты любых бизнес-процессов: действия, которые необходимо предпринять, способы их осуществления и контроля, требующиеся для этого ресурсы, а также визуализировать получаемые от этих действий результаты. CA ERwin Process Modeler повышает бизнес-эффективность ИТ-решений, позволяя аналитикам и проектировщикам моделей соотносить корпоративные инициативы и задачи с бизнес-требованиями и процессами информационной архитектуры и проектирования приложений. Таким образом, формируется целостная картина деятельности предприятия: от потоков работ в небольших подразделениях до сложных организационных функций.

CA ERwin Process Modeler эффективен в проектах, связанных с описанием действующих баз предприятий, реорганизацией бизнес-процессов, внедрением корпоративной информационной системы. Продукт позволяет оптимизировать деятельность предприятия и проверить ее на соответствие стандартам ISO 9000, спроектировать оргструктуру, снизить издержки, исключить ненужные операции и повысить эффективность. CA ERwin Process Modeler поддерживает методологии IDEF0 (функциональная модель), IDEF3 (Workflow Diagram) и DFD (Dataflow Diagram). Функциональная модель предназначена для описания существующих бизнес-процессов на предприятии (так называемая модель «как есть» – AS-IS) и идеального положения вещей – того, к чему нужно стремиться (модель «как должно

быть» – TO-BE). Методология IDEF0 предписывает построение иерархической системы диаграмм – единичных описаний фрагментов системы. Сначала проводится описание системы в целом и ее взаимодействия с окружающим миром (контекстная диаграмма), после чего проводится функциональная декомпозиция – система разбивается на подсистемы и каждая подсистема описывается отдельно (диаграммы декомпозиции). Затем каждая подсистема разбивается на более мелкие и так далее до достижения нужной степени подробности. После каждого сеанса декомпозиции проводится сеанс экспертизы: каждая диаграмма проверяется экспертами предметной области, представителями заказчика, людьми, непосредственно участвующими в бизнес-процессе. Такая технология создания модели позволяет построить модель, адекватную предметной области на всех уровнях абстракции.

* 1. Назначение CA ERwin Process Modeler

Пакет CA ERwin Process Modeler предназначен для создания функциональной модели существующей или проектируемой информационной системы. Функциональная модель включает в себя:

* поименованные процессы, функции или задачи, которые должны выполняться в системе;
* взаимодействие этих процессов, функций, задач с внешним миром и между собой.

CA ERwin Process Modeler 7.3 с использованием методологии IDEF0 позволяет наглядно представить выбранную систему как совокупность взаимодействующих функций и задач. Функции и задачи системы анализируются независимо от объектов, которыми они оперируют. Это позволяет более четко смоделировать логику и взаимодействие процессов организации.

* 1. Описание интерфейса CA ERwin Process Modeler

После запуска программы CA ERwin Process Modeler на экране появится окно программы (рисунок 1.1).



Рисунок 1.1 – Окно программы

Для создания новой модели необходимо вызвать диалог File → New или нажать на соответствующий значок на панели инструментов. После этого возникнет диалоговое окно, в котором следует указать название модели, выбрать методологию моделирования и нажать OK (рисунок 1.2).



Рисунок 1.2 – Окно создания новой модели

Далее появится окно, где следует указать свойства создаваемой модели (рисунок 1.3). На первой вкладке следует указать фамилию и имя автора модели, а также его инициалы. Остальные вкладки, определяющие такие свойства модели как: нумерация и положение функциональных блоков, высота и ширина страницы рекомендуется оставить без изменения.



Рисунок 1.3 – Окно свойств новой модели

На появившейся странице верхнего уровня модели находится первый функциональный блок модели (рисунок 1.4).

Рисунок 1.4 – Основное окно CA ERwin Process Modeler Основное окно программы содержит следующие части:

1. Рабочая область.
2. Панели инструментов.
3. Область модели.

Рассмотрим подробнее содержимое каждой из частей программы:

Рабочая область – содержит собственно разрабатываемую модель. На каждой странице отображается соответствующий уровень декомпозиции функциональной модели.

Панели инструментов: эти панели содержат практически все используемые при работе элементы. По умолчанию все панели отображаются на экране. При необходимости пользователь может отключить или, наоборот, включить требуемые модели, используя меню «View». Имеются следующие панели инструментов:

* Standard toolbar – содержит кнопки для управления файлами (новый, открыть, сохранить, печать), кнопки

отображения области свойств модели, кнопки управления масштабом изображения (рисунок 1.5).



Рисунок 1.5 – Стандартная панель инструментов

* Model toolbox – инструментальные кнопки создания элементов модели (рисунок 1.6).



Рисунок 1.6 – Панель Model toolbox

* Services Toolbar – панель кнопок специального инструментального средства, предназначенного для связывания пакета CA ERwin Process Modeler и пакета CA ERwin Data Modeler1 (рисунок 1.7).



Рисунок 1.7 – Панель Services Toolbar

Область модели содержит название модели, все уровни декомпозиции разрабатываемой функциональной модели, а также названия всех функций, выполняемых на каждом уровне декомпозиции.

1 CA ERwin Data Modeler  CASE-средство для проектирования и документирования баз данных, которое позволяет создавать, документировать и сопровождать базы данных, хранилища и витрины данных.

# 2 МЕТОДОЛОГИЯ IDEF0

* 1. Работа c функциональными блоками

Методология IDEF0 используется для создания функциональной модели, отображающей структуру и функции системы, а также потоки информации и материальных объектов, связывающие функции.

Каждая функциональная модель начинается с верхней диаграммы, имеющей только один прямоугольник-бокс, называемой контекстной, описывающей высший уровень функционирования модели. Контекстная диаграмма представляет границы изучаемого процесса с учетом цели, границ и точки зрения. Прямоугольник-бокс называется функциональным блоком.

Блок описывает функцию. Типичный функциональный блок показан на рисунке 2.1. Внутри каждого блока помещается его имя и номер. Имя должно быть активным глаголом или глагольным оборотом, описывающим функцию. Номер блока размещается в правом нижнем углу. Номера блоков используются для их идентификации на диаграмме и в соответствующем тексте.



Рисунок 2.1 – Пример функционального блока

Для того чтобы написать название процесса, функции или задачи, выполняемой проектируемой системой нужно два раза щелкнуть мышкой по первому функциональному блоку с номером А0, появившемуся после создания новой модели. После чего появится диалоговое окно Activity Properties, где будет предложено написать название функционального блока (рисунок 2.2). При выборе названия функционального блока следует использовать глагол, либо глагольное существительное,

обозначающее действие, так как разработка функциональной модели, заключается в описании функций, которые должна выполнять проектируемая система, и связей между ними.



Рисунок 2.2 – Окно свойств функционального блока

Также в этом диалоговом окне можно установить вид, стиль и размер шрифта надписи функционального блока, используя вкладку Font.

Для функциональной декомпозиции этого блока следует перейти в область модели, встать на появившееся название первого функционального блока и вызвать меню по нажатию правой кнопки мышки (рисунок 2.3).



Рисунок 2.3 – Декомпозиция функционального блока

Выбрать пункт меню «Decompose», после чего появится диалоговое окно с предложением выбрать количество блоков, на которые вы будете декомпозировать данный функциональный блок (рисунок 2.4), либо нажать на значок . Допустимый интервал числа функций 2-8. Число блоков на диаграмме в дальнейшем можно будет изменить (добавить недостающие или удалить лишние).

Рисунок 2.4 – Выбор количества функциональных блоков После ввода количества блоков на экране появится

следующая страница, отражающая второй уровень декомпозиции

проектируемой системы (рисунок 2.5).

Рисунок 2.5 **–** Второй уровень декомпозиции Функциональные блоки второго уровня декомпозиции

имеют номера А1, А2, А3, А4, А5. При декомпозиции каждого из них блоки на следующем уровне декомпозиции будут иметь номера соответственно А11, А12, А21, А22, А31, А32 и т.д. в зависимости от номера декомпозируемого функционального блока.

* 1. Работа со стрелками

Взаимодействие работ с внешним миром и между собой описывается в виде стрелок, которые называются ICOM-связями. Стрелка формируется из одного или более отрезков прямых и наконечника на одном конце. Стрелки не представляют поток или последовательность событий, как в традиционных блок-схемах потоков или процессов. Они лишь показывают, какие данные или материальные объекты должны поступить на вход функции для того, чтобы эта функция могла выполняться, и именуются существительными, например, «Заготовка», «Изделие», «Заказ».

В IDEF0 различают пять типов стрелок (рисунок 2.6).



Рисунок 2.6 – Виды стрелок

Каждый тип стрелок подходит к определенной стороне прямоугольника, изображающего работу, или выходит из нее:

1. Вход (input) – материал или информация, которые используются или преобразуется работой для получения результата (выхода). Допускается, что работа может не иметь ни одной стрелки входа. Стрелка входа рисуется как входящая в левую грань работы. При описании технологических процессов (для этого и был придуман IDEF0) не возникает проблем определения входов. При моделировании информационных систем, когда стрелками являются не физические объекты, а данные, не все так очевидно. Например, при "Приеме пациента" карта пациента может быть и на входе и на выходе, между тем качество этих данных меняется. Другими словами, в нашем примере для того, чтобы оправдать свое назначение, стрелки входа и выхода должны быть точно определены с тем, чтобы указать на то, что данные действительно были переработаны, например, на выходе – "Заполненная карта пациента”. Очень часто сложно определять, являются ли данные входом или управлением. В этом случае подсказкой может служить то, перерабатываются (изменяются) ли данные в работе или нет. Если изменяются, то, скорее всего, это вход, если нет – управление.
2. Управление (Control) – правила, стратегии, процедуры

или стандарты, которыми руководствуется работа. Каждая работа должна иметь хотя бы одну стрелку управления. Стрелка управления рисуется как входящая в верхнюю грань работы. Управление влияет на работу, но не преобразуется работой. Если цель работы - изменить процедуру или стратегию, то такая процедура или стратегия будет для работы входом. В случае возникновения неопределенности в статусе стрелки (управление или контроль) рекомендуется рисовать стрелку управления.

1. Выход (Output) – материал или информация, которая производится работой. Каждая работа должна иметь хотя бы одну стрелку выхода. Работа без результата не имеет смысла и не должна моделироваться. Стрелка выхода рисуется как исходящая из правой грани работы.
2. Механизм (Mechanism) – ресурсы, которые выполняют работу, например персонал предприятия, станки, устройства и т. д. Стрелка механизма рисуется как входящая в нижнюю грань работы. По усмотрению аналитика стрелки механизма могут не изображаться в модели.
3. Вызов (Call) – специальная стрелка, указывающая на другую модель работы. Используется для слияния и разветвления моделей. Стрелка механизма рисуется как исходящая из нижней грани работы. Стрелка вызова используется для указания того, что некоторая работа выполняется за пределами моделируемой системы.

Стрелки на контекстной диаграмме служат для описания взаимодействия системы с окружающим миром. Они могут начинаться у границы диаграммы и заканчиваться у работы, или наоборот. Такие стрелки называются граничными.

Для внесения граничной стрелки входа надо:

1. Щелкнуть по кнопке с символом стрелки в палитре инструментов следует перенести курсор к левой стороне экрана, пока не появится начальная штриховая полоска.
2. Щелкнуть один раз по полоске (откуда выходит стрелка) и еще раз в левой части работы со стороны входа (где заканчивается стрелка).
3. Вернуться в палитру инструментов и выбрать опцию

редактирования стрелки.

1. Щелкнуть правой кнопкой мыши на линии стрелки, во всплывающем меню выбрать Name Editor и добавить имя стрелки в закладке Name диалога IDEF0 Arrow Properties.

Стрелки управления, выхода, механизма и выхода изображаются аналогично. Для рисования стрелки выхода, например, следует щелкнуть по кнопке с символом стрелки в палитре инструментов, щелкнуть в правой части работы со стороны выхода (где начинается стрелка), перенести курсор к правой стороне экрана, пока не появится начальная штриховая полоска, и щелкнуть один раз по штриховой полоске. Имена вновь внесенных стрелок автоматически заносятся в словарь (Arrow Dictionary).

Словарь стрелок редактируется при помощи специального редактора Arrow Dictionary Editor, в котором определяется стрелка и вносится относящийся к ней комментарий. Словарь стрелок решает очень важную задачу. Диаграммы создаются аналитиком для того, чтобы провести сеанс экспертизы, т. е. обсудить диаграмму со специалистом предметной области. В любой предметной области формируется профессиональный жаргон, причем очень часто жаргонные выражения имеют нечеткий смысл и воспринимаются разными специалистами по- разному. В то же время аналитик – автор диаграмм должен употреблять те выражения, которые наиболее понятны экспертам. Поскольку формальные определения часто сложны для восприятия, аналитик вынужден употреблять профессиональный жаргон, а, чтобы не возникло неоднозначных трактовок, в словаре стрелок каждому понятию можно дать расширенное и, если это необходимо, формальное определение.

Содержимое словаря стрелок можно распечатать в виде отчета (меню Tools → Report → Arrow Report...) и получить тем самым толковый словарь терминов предметной области, использующихся в модели.

Несвязанные граничные стрелки (unconnected border arrow). При декомпозиции работы входящие в нее и исходящие из нее стрелки (кроме стрелки вызова) автоматически появляются на

диаграмме декомпозиции (миграция стрелок), но при этом не касаются работ. Такие стрелки называются несвязанными и воспринимаются в CA ERwin Process Modeler как синтаксическая ошибка.

На рисунке 2.7 приведен фрагмент диаграммы декомпозиции с несвязанными стрелками, генерирующийся CA ERwin Process Modeler при декомпозиции работы. Для связывания стрелок входа, управления или механизма необходимо перейти в режим редактирования стрелок, щелкнуть по наконечнику стрелки и щелкнуть по соответствующему сегменту работы. Для связывания стрелки выхода необходимо перейти в режим редактирования стрелок, щелкнуть по сегменту выхода работы и затем по стрелке.



Рисунок 2.7 – Пример несвязанных стрелок

Для связи работ между собой используются внутренние стрелки, т.е. стрелки, которые не касаются границы диаграммы, начинаются у одной и кончаются у другой работы.

Для рисования внутренней стрелки необходимо в режиме рисования стрелок щелкнуть по сегменту (например, выхода) одной работы и затем по сегменту (например, входа) другой. В IDEF0 различают пять типов связей работ.

Связь по входу (output-input), когда стрелка выхода вышестоящей работы (далее  просто выход) направляется на

вход нижестоящей, например, на рисунке 2.8 стрелка «Детали» связывает работы «Изготовление деталей» и «Сборка изделия».



Рисунок 2.8 – Связь по входу

Связь по управлению (output-control), когда выход вышестоящей работы направляется на управление нижестоящей. Связь по входу показывает доминирование вышестоящей работы. Данные или объекты выхода вышестоящей работы не меняются в нижестоящей работе. На рисунке 2.9 стрелка «Чертеж» связывает работы «Создание чертеже детали» и «Изготовление детали», при этом чертеж не претерпевает изменений в процессе изготовления деталей.



Рисунок 2.9 **–** Связь по управлению

Обратная связь по входу (output-input feedback), когда выход нижестоящей работы направляется на вход вышестоящей. Такая связь, как правило, используется для описания циклов. На рисунок 2.10 стрелка «Брак» связывает работы «Переработка сырья» и «Контроль качества», при этом выявленный на контроле брак направляется на вторичную переработку.



Рисунок 2.10 **–** Обратная связь по входу

Обратная связь по управлению (output-control feedback), когда выход нижестоящей работы направляется на управление вышестоящей (стрелка «Рекомендации», рисунок 2.11). Обратная связь по управлению часто свидетельствует об эффективности бизнес-процесса. На рисунок 2.11 качество изделия может быть повышено путем непосредственного регулирования процессами изготовления деталей и сборки изделия в зависимости от результата (выхода) работы «Контроль качества».



Рисунок 2.11 **–** Обратная связь по управлению

Связь выход-механизм (output-mechanism), когда выход одной работы направляется на механизм другой. Эта взаимосвязь используется реже остальных и показывает, что одна работа подготавливает ресурсы, необходимые для проведения другой работы (рисунок 2.12).



Рисунок 2.12 **–** Связь выход-механизм

Одни и те же данные или объекты, порожденные одной работой, могут использоваться сразу в нескольких других работах. С другой стороны, стрелки, порожденные в разных работах, могут представлять собой одинаковые или однородные данные или объекты, которые в дальнейшем используются или перерабатываются в одном месте. Для моделирования таких ситуаций в IDEF0 используются разветвляющиеся и сливающиеся стрелки. Для разветвления стрелки нужно в режиме редактирования стрелки щелкнуть по фрагменту стрелки и по соответствующему сегменту работы. Для слияния двух стрелок выхода нужно в режиме редактирования стрелки сначала щелкнуть по сегменту выхода работы, а затем по соответствующему фрагменту стрелки.

Смысл разветвляющихся и сливающихся стрелок передается именованием каждой ветви стрелок. Существуют определенные правила именования таких стрелок. Рассмотрим их на примере разветвляющихся стрелок. Если стрелка именована до разветвления, а после разветвления ни одна из ветвей не именована, то подразумевается, что каждая ветвь моделирует те же данные или объекты, что и ветвь до разветвления (рисунок 2.13).



Рисунок 2.13 **–** Пример именования разветвляющейся стрелки

Если стрелка именована до разветвления, а после разветвления какая-либо из ветвей не именована, то подразумевается, что эти ветви соответствуют именованию. Если при этом какая-либо ветвь после разветвления осталась неименованной, то подразумевается, что она моделирует те же данные или объекты, что и ветвь до разветвления (рисунок 2.14).



Рисунок 2.14 **–** Другой пример именования разветвляющейся стрелки

Недопустима ситуация, когда стрелка до разветвления не именована, а после разветвления не именована какая-либо из ветвей. CA ERwin Process Modeler определяет такую стрелку как синтаксическую ошибку (рисунок 2.15).



Рисунок 2.15 – Пример неверного именования разветвляющейся стрелки

Правила именования сливающихся стрелок полностью аналогичны, ошибкой будет считаться стрелка, которая после слияния не именована, а до слияния не именована какая-либо из се ветвей. Для именования отдельной ветви разветвляющихся и сливающихся стрелок следует выделить на диаграмме только одну ветвь, после этого вызвать редактор имени и присвоить имя стрелке. Это имя будет соответствовать только выделенной ветви.

Вновь внесенные граничные стрелки на диаграмме декомпозиции нижнего уровня изображаются в квадратных скобках и автоматически не появляются на диаграмме верхнего уровня (рисунок 2.16). Такие стрелки называются неразрешенными (unresolved).



Рисунок 2.16 – Неразрешенная стрелка

Для их «переноса» наверх нужно сначала выбрать кнопку на палитре инструментов щелкнуть по квадратным скобкам граничной стрелки. Появляется диалог Border Arrow Editor. Если щелкнуть по кнопке Resolve Border Arrow, стрелка мигрирует на диаграмму верхнего уровня, если по кнопке Change To Tunnel – стрелка будет затуннелирована и не попадет на другую

диаграмму. Тоннельная стрелка изображается с круглыми скобками на конце (рисунок 2.17).

Рисунок 2.17 **–** Типы туннелирования стрелок Туннелирование может быть применено для изображения

малозначимых стрелок. Если на какой-либо диаграмме нижнего уровня необходимо изобразить малозначимые данные или объекты, которые не обрабатываются или не используются работами на текущем уровне, то их необходимо направить на вышестоящий уровень (на родительскую диаграмму). Если эти данные не используются на родительской диаграмме, их нужно направить еще выше, и т.д. В результате малозначимая стрелка будет изображена на всех уровнях и затруднит чтение всех диаграмм, на которых она присутствует. Выходом является туннелирование стрелки на самом нижнем уровне. Такое туннелирование называется «не в родительской диаграмме».

Другим примером туннелирования может быть ситуация, когда стрелка механизма мигрирует с верхнего уровня на нижний, причем на нижнем уровне этот механизм используется одинаково во всех работах без исключения. (Предполагается, что не нужно детализировать стрелку механизма, т.е. стрелка механизма на дочерней работе именована до разветвления, а после разветвления ветви не имеют собственного имени). В этом случае стрелка механизма на нижнем уровне может быть удалена, после чего на родительской диаграмме она может быть затуннелирована, а в комментарии к стрелке или в словаре можно указать, что механизм будет использоваться во всех работах дочерней диаграммы декомпозиции. Такое туннелированние называется «не в дочерней диаграмме» (рисунок 2.17).

* 1. Проверка синтаксиса модели

Для проверки синтаксиса модели следует вызвать диалог Tools Reports → Model Consistency Report... После чего появится диалоговое окно (рисунок 2.18).

Затем следует выбрать пункт Preview для предварительного просмотра списка синтаксических ошибок модели. Список синтаксических ошибок может включать: неименованные функциональные блоки и стрелки (unnamed arrows, unnamed activities), несвязанные стрелки (unconnected border arrow), неразрешенные стрелки (unresolved (square tunneled) arrow connection), блоки, не имеющие, по крайней мере, одной стрелки выхода и одной стрелки управления (activity «Наименование функционального блока» has no Control) и т.д.



Рисунок 2.18 **–** Отчет проверки синтаксиса модели

Нажав на кнопку Print, Вы можете напечатать полученный отчет, кнопка Report позволяет сохранить сформированный отчет либо в формате txt , либо как файл Consistency Report, имеющий расширение bpc.

* 1. Формирование отчета Node Tree

Для наглядного представления количества уровней декомпозиции и отношений между родительскими и дочерними диаграммами следует сформировать отчет Node Tree. Для этого нужно вызвать диалог Diagram → Add Node Tree... После чего появится диалоговое окно, где будет предложено название отчета (можно написать другое) – Node Tree Name, верхний уровень диаграммы, с которого следует начать строить отчет – Top level

activity, и выбрать количество уровней который будет иметь отчет – Number of levels (рисунок 2.19).



Рисунок 2.19 **–** Окно построения отчета Node Tree

При нажатии кнопки далее можно изменить, либо оставить прежними параметры отчета. Затем следует нажать кнопку готово и появится сформированный отчет. Отчет имеет древовидную структуру (рисунок 2.20).



Рисунок 2.20 – Пример отчета Node Tree

* 1. Лабораторная работа № 1. Создание контекстной диаграммы и диаграмм декомпозиции
		1. Аннотация

Цель работы – получить навыки создания и редактирования функциональных моделей в CA ERwin Process Modeler.

В качестве примера рассматривается деятельность компании, занимающейся сборкой и продажей настольных компьютеров и ноутбуков. Компания не производит комплектующих, а только собирает и тестирует компьютеры.

Основные процедуры в компании:

1. Продавцы принимают заказы клиентов.
2. Операторы группируют заказы по типам компьютеров.
3. Операторы собирают и тестируют компьютеры.
4. Операторы упаковывают компьютеры согласно заказам.
5. Кладовщик отгружает клиентам заказы.

Компания использует купленную бухгалтерскую информационную систему, которая позволяет оформить заказ, счет и отследить платежи по счетам.

* + 1. Ход работы
1. Запустите CA ERwin Process Modeler. Автоматически будет вызван диалог для создания модели.
2. Запустите CA ERwin Process Modeler. Автоматически будет вызван диалог для создания модели (В случае если диалоговое окно автоматически вызвано не было выполните команду меню File → New…).
3. Выберите тип модели IDEF0 и назовите ее

«Деятельность компании» (рисунок 2.21).



Рисунок 2.21 – Диалоговое окно создания модели в CA ERwin

Process Modeler

1. Для отображения / скрытия инструментов просмотра и навигации используйте команду меню View → Model Explorer…
2. Перейдите в меню Model → Model Properties... На вкладке General внесите имя модели (Model name) «Деятельность компании», название проекта (Project name) «Моделирование деятельности компании», имя автора (Author) и тип модели (Time Frame) – «модель как есть» (AS-IS) (рисунок 2.22).



Рисунок 2.22 – Диалоговое окно свойств модели

1. На вкладке Purpose внесите цель диаграммы (Purpose) –

«Цель: Моделировать текущие бизнес-процессы компании» и точку зрения на модель (Viewpoint) – «Точка зрения: Директор».

1. На вкладке Definition внесите описание модели (Definition) – «Это «модель как есть», описывающая деятельность компании по сборки и тестированию компьютеров» и область применения модели (Scope) – «Общее управление бизнесом компании: исследование рынка, закупка комплектующих, сборка, тестирование и продажа компьютеров».
2. Закройте диалоговое окно создания модели (нажмите кнопку ОК) и вернитесь на контекстную диаграмму.
3. Щелкнув правой кнопкой мыши по функциональному блоку А0, вызовите контекстное меню и выберите пункт Name

(альтернативный способ – левой кнопкой мыши дважды щелкните по функциональному блоку А0). В появившемся диалоговом окне свойств функционального блока (Activity Properties) на вкладке Name (рисунок 2.23) задайте название диаграммы «Деятельность компании». На вкладке Definition внесите описание функционального блока «Текущие бизнес- процессы компании».



Рисунок 2.23 – Диалоговое окно свойств функционального блока

1. С помощью инструмента для создания стрелок (Precedance Arrow Tool)  панели Model Toolbox создайте на контекстной диаграмме граничные стрелки, представленные в таблице 2.1. Название и описание стрелок задается по аналогии с функциональными блоками.

Таблица 2.1 – Граничные стрелки контекстной диаграммы

|  |  |  |
| --- | --- | --- |
| Название | Описание | Тип |
| Бухгалтерская система | Оформление счетов, оплатасчетов, работа с заказами | Механизм |
| Звонки клиентов | Запросы информации, заказы,техподдержка и т.д. | Вход |
| Маркетинговые материалы | Рекламные материалы, прайс-листы | Выход |
| Правила и процедуры | Правила продаж, инструкции посборке, процедуры тестирования, критерии производительности и др. | Управление |
| Проданные продукты | Настольные компьютеры иноутбуки | Выход |

1. С помощью инструмента для управления текстовыми блоками (Text Tool)  панели Model Toolbox внесите на диаграмму текст цели и точки зрения. Результатом выполнения последних пунктов должна стать контекстная диаграмма, представленная на рисунке 2.24.



Рисунок 2.24 – Конкретная диаграмма

1. Создайте отчет по модели. Для этого выполните команду меню Tools → Reports → Model Report…, установите все флажки в диалоговом окне настройки отчета по модели (Model

Report), как показано на рисунке 2.25, и нажмите кнопку Preview...



Рисунок 2.25 – Диалоговое окно настройки отчета по модели

1. Вернитесь на контекстную диаграмму. Нажмите кнопку перехода на диаграмму декомпозиции (Go to Child Diagram) . Автоматически будет вызван диалог для создания диаграммы декомпозиции для контекстной диаграммы (рисунок 2.26). Укажите количество дочерних работ равное трем.



Рисунок 2.26 – Диалоговое окно создания диаграммы декомпозиции

1. Перейдите на дочернюю диаграмму. Откройте диалоговое окно свойств каждого функционального блока и укажите в поле Name имя функционального блока, а в поле Description – ее описание согласно таблице 2.2.

Таблица 2.2 – Функциональные блоки диаграммы А0

|  |  |  |
| --- | --- | --- |
| Номер | Название | Описание |
| А1 | Продажи и маркетинг | Телемаркетинг, презентации и выставки |
| А2 | Сборка и тестированиекомпьютеров | Сборка и тестирование настольныхкомпьютеров и ноутбуков |
| А3 | Хранение и отгрузка | Получение комплектующих, хранение компьютеров и комплектующих,отгрузка заказов клиентам |

1. Используя инструмент для создания стрелок (Precedance Arrow Tool)  панели Model Toolbox, свяжите граничные стрелки так: как показано на рисунке 2.27.



Рисунок 2.27 – Связанные граничные стрелки на диаграмме А0

1. Откройте диалоговое окно свойств ветви стрелки управления (Arrow Properties) для работы А2 и переименуйте ее в

«правила сборки и тестирования».

1. Аналогично переименуйте стрелку механизма для работы А1 в «Систему оформления заказов».
2. Создайте стрелки связи по входу: между функциональными блоками А1 и А2 – «Заказы клиентов»; между функциональными блоками А2 и А3 – «Собранные компьютеры».
3. Создайте стрелку обратной связи по управлению между функциональными блоками А2 и А1 – «Результаты сборки и тестирования».
4. Правой кнопкой мыши щелкните на созданную стрелку

«Результаты сборки и тестирования», выберите пункт контекстного меню Style… (Стиль) и установите большую толщину стрелки. Также в контекстном меню установите опцию Extra Arrowhead (Дополнительный указатель стрелки) для более наглядного отображения этой стрелки.

1. Разместите названия стрелок таким образом, чтобы их было удобно читать и их принадлежность к стрелке трактовалась однозначно. При необходимости установите опцию Squiggle (Тильда) из контекстного меню.
2. Результат проведенных действий представлен на рисунке 2.28.



Рисунок 2.28 – Диаграмма А0

1. В результате проведения экспертизы была получена следующая информация относительно функционального блока

«Сборка и тестирование компьютеров»:

* Производственный отдел получает заказы клиентов от отдела продаж по мере их поступления.
* Диспетчер координирует работу сборщиков, сортирует заказы, группирует их и дает указание на отгрузку компьютеров, когда они готовы.
* Каждые два часа диспетчер группирует заказы – отдельно для настольных компьютеров и ноутбуков – и направляет на участок сборки.
* Сотрудники участка сборки собирают компьютеры согласно спецификациям заказа и инструкциям по сборке. Когда группа компьютеров, соответствующая группе заказов, собрана, она направляется на тестирование.
* Тестировщики тестируют каждый компьютер и в случае необходимости заменяют неисправные комплектующих.
* Тестироващики направляют результаты тестирования диспетчеру, который на основании этой информации принимает решение о передаче компьютеров, соответствующей группе заказов, на отгрузку.
1. На основании полученной информации необходимо создать диаграмму декомпозиции работы А2. Для этого выделите эту работу, нажмите кнопку перехода на диаграмму декомпозиции (Go to Child Diagram)  и укажите количество дочерних функциональных блоков равное четырем.
2. Назовите функциональные блоки и внесите их описание согласно данным таблицы 2.3.

Таблица 2.3 – Функциональные блоки диаграммы А2

|  |  |  |
| --- | --- | --- |
| Номер | Название | Описание |
| А21 | Отслеживание расписания и управление сборкой итестирование | Просмотр заказов, установка расписания выполнения заказов, просмотррезультатов тестирования, формирование групп заказов на сборку и отгрузку |
| А22 | Сборка настольных компьютеров | Сборка настольных компьютеров всоответствии с инструкциями и указаниями диспетчера |
| А23 | Сборка ноутбуков | Сборка ноутбуков в соответствии синструкциями и указаниями диспетчера |
| А24 | Тестированиекомпьютеров | Тестирование компьютеров и компонентов.Замена не работающих комплектующих |

1. Создайте на диаграмме А2 стрелки в соответствии с таблицей 2.4.

Таблица 2.4 – Стрелки диаграмма А2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Название | Источник | Тип | Назначение | Тип |
| Заказы клиентов | Границадиаграммы | Управление | А21 | Управление |
| Заказы на настольныекомпьютеры | А21 | Выход | А22 | Управление |
| Заказы на ноутбуки | А21 | Выход | А23 | Управление |
| Комплектующие | Границадиаграммы | Вход | А22, А23,А24 | Вход |
| Настольныекомпьютеры | А22 | Выход | А24 | Вход |
| Ноутбуки | А23 | Выход | А24 | Вход |
| Персоналпроизводственного отдела | Граница диаграммы | Механизм | А22, А23 | Механизм |
| Правила сборки итестирования | Границадиаграммы | Управление | А22, А23,А24 | Управление |
| Результаты сборки итестирования | А22, А23,А24 | Выход | Границадиаграммы | Выход |
| Собранныекомпьютеры | А24 | Выход | Границадиаграммы | Выход |
| Тестировщик | Расщепление | Механизм | А24 | Механизм |
| Диспетчер | Расщепление | Механизм | А21 | Механизм |
| Указание передать компьютеры наотгрузку | А21 | Выход | А24 | Управление |

1. При создании стрелок следует учесть, что стрелки

«Комплектующие» и «Персонал производственного отдела» будут туннелями типа «не в родительской диаграмме».

1. Источник «Расщепление» означает, что стрелка является именованной ветвью другой стрелки диаграммы.
2. Результат создания функциональных блоков и стрелок приведен на рисунке 2.29



Рисунок 2.29 – Диаграмма А2

1. По завершению построения диаграммы необходимо проверить синтаксис модели.
	1. Лабораторная работа № 2. Создание диаграммы «Дерево узлов» и диаграммы «Только для экспозиции»
		1. Аннотация

Цель работы – получить навыки создания и редактирования диаграммы «Дерево узлов» и диаграммы «Только для экспозиции».

Диаграмма «Дерево узлов» (Node Tree) – иерархическая диаграмма IDEF0-модели или часть модели, которая иллюстрирует функции и связи (родительская – дочерняя) между ними. Дерево узлов обеспечивает обзор модели в целом. Верхний узел в иерархии соответствует функции контекстной диаграммы и уровню дочерних декомпозиций контекстной функции, составляющей основу дерева.

Диаграмма «Только для экспозиции» (FEO, For Exposition Only) используется как средство объяснения части процесса, чтобы показать другую точку зрения или выделить другие функциональные детали, что явно не поддерживается синтаксисом IDEF0. Диаграмма «Только для экспозиции» может быть аннотирована дополнительным пояснительным текстом и может не следовать правилам IDEF0.

Диаграммы «Только для экспозиции» используют:

* как инструмент проработки вариантов стандартной диаграммы, в случае, если нет уверенности в правильности или оптимальности разработки;
* как возможность создания упрощенных диаграмм, которые позволяют акцентировать внимание на особенностях диаграммы или облегчают ее понимание;
* как возможность разделения диаграмм на отдельные фрагменты, осуществляя эту декомпозицию не в правилах IDEF0.

Отличия в работе с диаграммами «Только для экспозиции»:

* функциональные блоки диаграммы «Только для экспозиции» не могут быть декомпозированы;
* миграция граничных стрелок не может происходить при независимости диаграммы «Только для экспозиции» от ее родительской диаграммы, сохраняется как

исключение двусторонняя зависимость по имени функционального блока;

* с затуннелированными на диаграмме «Только для экспозиции» стрелками возможны манипуляции в как IDEF0, но стрелки мигрировать не могут.

В качестве примера диаграммы «Только для эксплозии» рассматривается ранее созданная модель деятельности компании, занимающейся сборкой и продажей настольных компьютеров и ноутбуков с оговоркой, что сборкой настольных компьютеров и ноутбуков занимается один отдел.

* + 1. Ход работы
1. Выполните команду меню Diagram → Add Node Tree….
2. В первом диалоговом окне мастера создания диаграммы

«Дерево узлов» (Node Tree Wizard) внесите имя диаграммы, укажите корневую диаграмму дерева узлов и количество его уровней, как представлено на рисунке 2.30.



Рисунок 2.30 – Диалоговое окно мастера создания диаграммы

«Дерево узлов». Шаг 1

1. На втором шаге мастера создания диаграммы «Дерево узлов» установите параметры прорисовки диаграммы «Дерево узлов» так, как показано на рисунке 2.31.



Рисунок 2.31 – Диалоговое окно мастера создания диаграммы

«Дерево узлов». Шаг 2

1. Результат работы мастера создания диаграммы «Дерево узлов» являются диаграмма, представленная на рисунке 2.32.



Рисунок 2.32 – Диаграмма «Дерево узлов»

1. Диаграмму «Дерево узлов» можно модифицировать. Например, нижний уровень может быть отображен в виде прямоугольников, так же как и верхние уровни, а соединяющие функциональные блоки линии могут быть не только ортогональными, но и диагональными. Для этого создайте еще одну диаграмму «Дерево узлов» с именем «Деятельность компании (модифицированная)». Только на этот раз на втором шаге мастера создания диаграммы «Дерево узлов» в диалоговом окне в группе свойств рисунка (Drawing) отключите свойства

«Маркер последнего уровня» (Bullet Last Level), а в группе свойств стиля соединений (Connection Style) переключите отображения с ортогональные линии (Orthogonal lines) на диагональные линии (Diagonal lines). Результат представлен на рисунке 2.33.



Рисунок 2.33 – Модифицированная диаграмма «Дерево узлов»

1. Для создания альтернативного представления процесса сборки и тестирования компьютеров выполните пункт меню

«Diagram → Add FEO diagram…»

1. В появившемся диалоге выберите тип диаграммы – диаграмма декомпозиции (Decomposition Diagram), назначьте источником (Source Diagram Name) диаграмму А2 и задайте имя создаваемой диаграмме «Только для экспозиции», как представлено на рисунке 2.34.



Рисунок 2.34 – Диалоговое окно создания диаграммы «Только для экспозиции»

1. Выполните команду меню Diagram → Diagram Properties.... В появившемся диалоге выберите закладку

«Описание диаграммы» (Diagram Text) и внесите описание диаграммы согласно рисунку 2.35.



Рисунок 2.35 – Описание диаграммы «Только для экспозиции»

1. Удалите функциональные блоки А22 и А23.
2. Добавьте функциональный блок «Сборка настольных компьютеров и ноутбуков».
3. Создайте стрелку связи по управлению между функциональными блоками А21 и А22 – «Заказы на настольные компьютеры и ноутбуки».
4. Создайте стрелку связи по входу между функциональными блоками А22 и А23 – «Настольные компьютеры и ноутбуки».
5. Создайте граничную связь выхода из функционального блока А22 – «Результаты сборки и тестирования».
6. Удалите лишние стрелки. Результат должен соответствовать рисунку 2.36.



Рисунок 2.36 – Диаграмма «Только для экспозиции»

1. Для перехода между диаграммой IDEF0, деревом узлов и FEO-диаграммой используйте кнопку «Перейти на родственную диаграмму» (Go to Sibling Diagram) .
	1. Лабораторная работа № 3. Функционально-стоимостное моделирование
		1. Аннотация

Цель работы – получить навыки функционально- стоимостного моделирования с использованием технологии

«Стоимость, основанная на функциях».

Функциональную-стоимостное моделирование основывается на постатейном распределении параметризованных (оцифрованных) затрат времени и финансовых ресурсов, связанных с осуществлением тех или иных функций.

В CA ERwin Process Modeler функционально-стоимостное моделирование основывается на технике для сбора и анализа стоимости функций, называемая «Стоимость, основанная на функциях» (Activity-Based Costing, ABC).

Технология предполагает, что сначала выявляются центры затрат (Cost Center), после чего они связываются с функциональными блоками. При этом родительские функциональные блоки обычно консолидируют стоимости связанных с ними дочерних функциональных блоков.

В качестве примера для функционально-стоимостного моделирования рассматривается ранее созданная модель деятельности компании, занимающейся сборкой и продажей настольных компьютеров и ноутбуков.

* + 1. Ход работы
1. Выполните команду меню Model → Model Properties…, перейдите на вкладку ABC Units и установите единицы измерения денег и времени – российский рубль и часы (Hours), как представлено на рисунке 2.5.



Рисунок 2.37 – Настройка единиц измерения для функционально- стоимостного моделирования

1. Перейдите в словарь центров затрат (команда меню Dictionary → Cost Center…) и внесите названия и описание центров затрат согласно таблице 2.5.

Таблица 2.5 – Центры затрат

|  |  |
| --- | --- |
| Название | Описание |
| Заработная плата | Затраты на оплату труда рабочих, занятых сборкой и тестированием компьютеров, и диспетчера, занятого составлением графика работ, формированием партийкомпьютеров, контролем над сборкой и тестированием |
| Отчисления сзаработной платы | Затраты на уплату страховых взносах во внебюджетные фонды. Страховая нагрузка составляет 30% отзаработной платы |
| Комплектующие | Затраты на закупку комплектующих |

1. Для того чтобы стоимость процессов отображалась на диаграммах, выполните команду меню Model → Model Properties..., перейдите на закладку «Отображение» (Display), установите опцию ABC Data и выберите отображение затрат (Cost), как показано на рисунке 2.38



Рисунок 2.38 – Настройка отображения данных о функционально-стоимостном моделирование

1. Для назначения стоимости процессов следует щелкнуть правой кнопкой мыши по функциональному блоку и выбрать в контекстном меню пункт «Затраты» (Cost). В появившемся диалоге необходимо заполнить затраты текущего процессу по каждому центру затрат, указать его частоту (Frequency) и длительность (Duration), как показано на рисунке 2.39.



Рисунок 2.39 – Заполнение затрат на процесс «Отслеживание расписания и управление сборкой и тестированием»

1. Заполните затраты по функциональным блокам на диаграмме А2 согласно таблице 2.6.

Таблица 2.6 – Стоимость процессов на диаграмме А2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Функциональныйблок | Центр затрат | Стоимость | Длительность | Частота |
| Отслеживание расписания и управление сборкойи тестированием | Заработная плата | 2 000,00 | 8,0 | 3 |
| Отчисления сзаработной платы | 6 00,00 |
| Сборка настольных компьютеров | Заработная плата | 1 000,00 | 0,5 | 20 |
| Комплектующие | 10 000,00 |
| Отчисления сзаработной платы | 300,00 |
| Сборка ноутбуков | Заработная плата | 2 000,00 | 1,0 | 12 |
| Комплектующие | 15 000,00 |
| Отчисления сзаработной платы | 600,00 |
| Тестирование компьютеров | Заработная плата | 400,00 | 0,3 | 32 |
| Отчисления сзаработной платы | 120,00 |

1. Убедитесь в том, что на диаграмме более высокого уровня была рассчитана стоимость процесса «Сборка и тестирование компьютера» (рисунок 2.40).



Рисунок 2.40 – Отображения стоимости на функциональном блоке «Сборка и тестирование компьютеров»

1. Сгенерируйте отчет о стоимостных затратах на процесс

«Сборка и тестирование компьютеров». Для этого выполните команду меню Tools → Reports → Activity Cost Report…. Заполните диалог так, как показано на рисунке 2.41 и нажмите кнопку «Предварительный просмотр» (Preview).



Рисунок 2.41 – Генерация отчета по функционально- стоимостному моделированию

* 1. Лабораторная работа № 4. Анализ пользовательских свойств
		1. Аннотация

Цель работы – получить навыки связывания различной специфической бизнес-информации с функциональными блоками и связями между ними.

Если стоимостных показателей недостаточно для анализа деятельности организации, то имеется возможность внесения собственных метрик – определенных пользователем свойств (User Defined Properties, UDP).

Работа с UDP начинается с создания словаря UDP (UDP Dictionary) и словаря ключевых слов UDP (UDP Keywords Dictionary). Разработанные словари являются основой для присвоения объектам функциональных диаграмм характеристик в пределах понятий словаря UDP. В дальнейшем эти характеристики подлежат обзору и анализу.

В качестве примера для анализа пользовательских свойств рассматривается ранее созданная модель деятельности компании, занимающейся сборкой и продажей настольных компьютеров и ноутбуков.

4.8.2 Ход работы

1. Выполните команду меню Dictionary → UDP Keywords и внесите следующие ключевые слова (рисунок 2.42):
	* Расход ресурсов.
	* Документация.
	* Информационная система.



Рисунок 2.42 – Ключевые слова UDP

1. Выполните команду Dictionary → UDP… и внесите в словарь название и описание пользовательских свойств в таблице 2.7.

Таблица 2.7 – Описание определенных пользователей свойств в словаре

|  |  |  |  |
| --- | --- | --- | --- |
| Название UDP | Тип | Значение | Ключевоеслово UDP |
| Дополнительная документация | Command List | powerpnt.exe sample1.ppt winword.exe sample1.docwinword.exe sample2.doc | Документация |
| Загрязнение окружающей среды | Text List (Single selection) | Очень высокое ВысокоеСреднее НизкоеОчень низкое |  |
| Историяизменений | Paragraph Text |  | Документация |
| Приложения | Text List (Multiple selection) | Модуль оформления заказовМодуль процедур сборки и поиска неисправностей Модель создания иконтроля расписания | Информационн ая система |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | работ Модуль учетакомплектующих и оборудования |  |
| Расходэлектроэнергии | Real Number |  | Расходресурсов |

1. Для назначения UDP функциональному блоку нужно щелкнуть по ней правой кнопкой мыши и выбрать пункт контекстного меню UDP…. В появившемся диалоге следует указать необходимые свойства (рисунок 2.43).



Рисунок 2.43 – Установка пользовательских свойств для функциональных блоков

1. Для диаграммы декомпозиции А2 внесите значения пользовательских свойств согласно таблицы 2.8.

Таблица 2.8 – Значения пользовательских свойств для диаграмм А2

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Функцио- нальный блок | Дополнитель- наядокументация | Приложения | История изменений | Расход электро- энергии | Загрязне- ние окружаю- щейсреды |
| А21 | winword.exe sample1.doc | Модульсоздания и контроля расписанияработ | История изменений специфика ций | 10 | Низкое |
| А22 |  | Модуль учета комплектующих и оборудования Модуль процедур сборки и поисканеисправностей |  | 20 | Среднее |
| А23 |  | Модуль учета комплектующих и оборудования Модуль процедур сборки и поисканеисправностей |  | 25 | Среднее |
| А24 |  | Модуль учета комплектующих и оборудования Модуль процедур сборки и поисканеисправностей |  | 40 | Среднее |

1. После внесения пользовательских свойств типа

«Командная строка» (Command), проверьте правильность выполнения команды, для чего нажмите на кнопку .

1. Убедитесь в том, что на функциональных блоках, которые обладают пользовательскими свойствами в правом верхнем углу прямоугольника появилась иконка . Двойной щелчок левой кнопкой мышки по иконке вызывает диалоговое окно UDP для этого функционального блока.
2. Откройте диалог UDP для работы «Сборка настольных компьютеров» и нажмите кнопку «Фильтр» (Filter). В появившемся окне отключите ключевое слово «Информационная система», как показано на рисунке 2.44.



Рисунок 2.44 – Фильтрация UDP-свойств для работы

1. Создайте отчет по UDP. Для этого выполните команду меню Tools → Reports → Diagram Object Report и выберите следующие опции (см. рисунок 2.45):
	* Start From Activity: «А2: Сборка настольных компьютеров».
	* Number of Levels: 2.
	* User-Defined Properties: «Расход электроэнергии».



Рисунок 2.45 – Создание отчета UDP

3 МЕТОДОЛОГИЯ IDEF3

* 1. Основные элементы методологии IDEF3

Методология IDEF3 предназначена для описания потоков работ (Workflow Modeling). IDEF3 широко используется для создания моделей бизнес-процессов организации на нижнем уровне – при описании работ, выполняемых в подразделениях и на рабочих местах2.

Основным графическим объектом моделей методологии IDEF3, являются единицы работы (Unit of Work, UOW)3. Они служат для описания функций (работ, процессов), выполняемых подразделениями/сотрудниками предприятия. Обозначение единицы работы на моделях IDEF3 представлено на рисунке 3.1.



Рисунок 3.1 – Единица работы (Unit of Work, UOW)

Другой, не менее важный, тип объектов – стрелки. Они используются либо для отражения в модели последовательности выполнения функций во времени (стрелка предшествования), либо последовательности выполнения функций, обусловленной потоком материальных объектов (ресурсов) (стрелка потока объектов). Обозначения стрелок на моделях IDEF3 представлено на рисунке 3.2.

1. Нотация IDEF3 взята за основу при создании методологии описания процессов ARIS eEPC
2. В тексте IDEF3 для обозначения этого оператора используется другой термин – единица поведения (Unit of Behavior, UOB)

 

а) б)

Рисунок 3.2 – Стрелки в методологии IDEF3 (а – стрелка предшествования, б – стрелка потока объектов)

Методология IDEF3 не имеет какой-либо регламентации функционального назначения стрелок-связей, кроме как «вход» и

«выход». Отсутствуют также какие-либо топологические ограничения.

Особенности этой методологии определяются в первую очередь, тем, что она отражает функции системы в причинной последовательности их осуществления. Рассматриваются параллельные или альтернативные процессы.

Это отличие вызывает периодическую необходимость решения вопросов о выборе возможных вариантов последовательности действий или координации этих действий. Такая необходимость возникает всякий раз, когда связи, символизирующие в IDEF3 условия начала осуществления функции. Разветвляются или сливаются, что приводит к использованию при моделировании в IDEF3 нового оператора –

«перекресток» (Junction).

Ведение в IDEF3 перекрестка является обязательным условием каждого слияния и каждого разветвления стрелок. Поэтому каждый из перекрёстков может быть использован в двух вариантах:

* в качестве перекрестка слияния (Fan-in);
* в качестве перекрестка разветвления (Fan-out).

Кроме того, все перекрестки, кроме перекрестка XOR (эксклюзивное ИЛИ), характеризуются как:

* синхронные (Synchronous) перекрестки;
* асинхронные (Asynchronous) перекрестки.

Эта классификация предопределяет временные соотношения совершения событий, предшествующих перекрёстку (для перекрестков слияния), и событий, последующих за перекрёстком (для перекрёстков разветвления).

В таблице 3.1 приведен перечень операторов «Перекресток» и пояснения к ним.

Таблица 3.1 – Перекрестки в методологии IDEF34

|  |  |  |
| --- | --- | --- |
| Перекресток | Перекресток слияния(Fan-in) | Перекрестокразветвления (Fan-out) |
| Асинхронный AND | Все предшествующие процессы должны бытьзакончены | Все последующие процессы должны бытьначаты |
| Синхронный AND | Все предшествующие процессы должны быть законченыодновременны | Все последующие процессы должны быть начаты одновременно |
| Асинхронный OR | Один или больше предшествующих процессов должны бытьзакончены | Один или большепоследующих процессов должны быть начаты |
| Синхронный OR | Один или больше предшествующих процессов должны быть законченыодновременно | Один или большепоследующих процессов должны быть начаты одновременно |
| Эксклюзивный OR | Только один предшествующий процесс должен бытьзакончен | Только одинпоследующий процесс должен быть начат |

Еще одним элементом моделей IDEF3 является внешняя ссылка (Referent). Объект «внешняя ссылка» используется для описания ссылок на другие диаграммы модели, циклические переходы в рамках одной модели, различные комментарии к функциям и перекресткам.

На рисунке 3.3 представлено обозначение объекта «внешняя ссылка» в методологии IDEF3.

1. Сущность оператора «Перекресток ИЛИ» подобна сущности оператора «Решение» в ГОСТ 19.701-90. Этот оператор осуществляет функцию графического отражения логического выбора между альтернативными вариантами функционирования.



Рисунок 3.3 – Внешняя сущность (Referent)

Как и функциональные блоки методологии IDEF0 единицы работы методологии IDEF3 сопровождаются текстовыми комментариями, вводимыми в информационные поля окна UOW:

* + Объекты (Objects) – детализирующий раздел IDEF3, который отражает физические объекты, участвующие в процессе UOW. Описания объектов для UOW должны определить, является ли объект продуктом процесса, преобразуется процессом, участвует вне процесса, преобразуется, создает или уничтожается в течение процесса.
	+ Факты (Facts) – детализирующий раздел IDEF3, который перечисляет утверждения об UOW или объекты, причастные к UOW, включая свойства объекта и связи, которые поддерживаются между объектами в течение процесса. Факты об UOW могут также включить свойства UOW, такие как длительность, частота или стоимость.
	+ Описание (Description) – детализирующий раздел IDEF3, который содержит текстовое описание UOW, которое затем используется как словарный вход для UOW. Описание подробно излагает информацию в списках объектов, фактов и ограничений.
	+ Ограничения (Constraints) – детализирующий раздел IDEF3, который перечисляет ограничивающие UOW факторы, такие как, например условия, которые должны быть выполнены для осуществления запуска, продолжения и завершения процесса. Ограничения являются обычно фактами об UOW, которые связаны с ним или управляют его осуществлением, описывая показатели, влияющие на UOW перед, в течение, или

после его выполнения, или, что должно всегда присутствовать, или никогда не присутствовать при выполнении, чтобы оно произошло.

Методологию IDEF3 целесообразно применять в случае относительно простых процессов на нижнем уровне декомпозиции, то есть процессов уровня рабочих мест. В этом случае схема процесса может служить основой для создания документов, регламентирующих работу исполнителей. Очевидно, что процесс в методологии IDEF3 является «плоским». При помощи этой методологии достаточно сложно создавать комбинированные модели, в которых бы сочетались описания потоков работ и процессы управления этими работами.

* 1. Лабораторная работа № 5. Функциональное моделирование в методологии IDEF3
		1. Аннотация

Цель работы – получить навыки создания и редактирования моделей методологии IDEF3 в CA ERwin Process Modeler.

В качестве примера рассматривается бизнес-модель деятельности компании, построенная в предыдущих лабораторных работах.

* + 1. Ход работы
1. Перейдите на диаграмму А2 и декомпозируйте функциональны блок «Сборка настольных компьютеров». В диалоге декомпозиции установите нотацию IDEF3 и число работ 4 (рисунок 3.4)



Рисунок 3.4 – Декомпозиция функционального блока в методологии IDEF3

1. Щелкните правой кнопкой мыши по первой работе на диаграмме и выберите в контекстном меню команду Name (Имя).
2. Внесите название работы «Подготовка комплектующих». Перейдите на вкладку «Definition» (Определение) и внесите описание работы «Подготавливаются все комплектующие компьютера согласно спецификации заказа».
3. Перейдите на вкладку «UOW» и внесите следующие свойства работы:
	* Objects (Объекты): «Компоненты: HDD, SSD, корпуса, материнские платы, видеокарты, дисководы DVD, Blu-Ray и FDD, программное обеспечение».
	* Facts (Факты): «Доступные операционные системы: Windows 8.1, Windows 10, Ubuntu, ChromeOS».
	* Constrains (Ограничения): «Установка дискретной видеокарты требует установки дополнительного программного обеспечения (драйверов)».
4. С помощью кнопки установки блоков (Activity Box Tool)  необходимо добавить еще 10 работ:
	* Установка материнской платы.
	* Установка HDD.
	* Установка SSD.
	* Установка привода DVD.
	* Установка привода Blu-Ray.
	* Установка FDD.
	* Установка видеокарты.
	* Установка процессора и оперативной памяти.
	* Инсталляция операционной системы.
	* Инсталляция дополнительного программного обеспечения.
5. Используя инструмента добавления на модель объекта внешней ссылки (Referent Tool) , создайте объект ссылки и назовите его «Комплектующие».
6. Свяжите стрелкой созданный объект ссылки и работу

«Подготовка комплектующих».

1. Создайте объект ссылки «Программное обеспечение» и свяжите его с работами «Инсталляция операционной системы» и

«Инсталляция дополнительного программного обеспечения».

1. С помощью инструмента добавления перекрестка (Junction Tool)  создайте два перекрестка типа «асинхронное ИЛИ» и два перекрестка типа «исключающее ИЛИ».
2. Соедините работы и перекрестки между собой в соответствии с рисунком 3.5.



Рисунок 3.5 – Диаграмма IDEF3

1. Опишите принципы работы перекрестков на полученной диаграмме.
	1. Лабораторная работа № 6. Разработка диаграммы Swim Lane на основе ролей производственного подразделения
		1. Аннотация

Цель работы – получить навыки создания и редактирования диаграммы Swim Lane с временными шкалами на основе ролей производственного подразделения.

Swim Lane (Плавательная дорожка) – нестандартная диаграмма, демонстрирующая распределение функций IDEF3- диаграммы между исполнителями (Roles), организованными в производственные подразделения (Role Group).

Начиная разрабатывать Swim Lane, надо иметь в виду следующее:

* + Swim Lane может быть построена на основе диаграмм IDEF3.
	+ Условием построения Swim Lane является наличие словаря производственных подразделений (Role Group Dictionary).
	+ Как альтернатива Swim Lane может быть построена на основе пользовательских свойств с типом данных Single Select Text List.
	+ Для Swim Lane можно построить горизонтальные шкалы с различным предметным содержанием (время, стоимость и другие), определяющимся пользовательскими свойствами. В этом случае пользовательские свойства должны иметь типы данных из альтернатив: Integer или Real Number.
	+ Каждая Swim Lane диаграмма состоит из точно того же набора объектов, что и базовая диаграмма IDEF3. Внешнее различие вызвано лишь необходимостью распределения объектов по дорожкам диаграммы Swim Lane.
	+ Отличие диаграммы Swim Lane от стандартной диаграммы IDEF3 состоит в том, что ее рабочее поле разделено на несколько горизонтальных полос (Swim Lane), каждая из которых олицетворяет функции того или другого должностного лица (Role).
	+ Количество дорожек равно количеству ролей (Roles) в составе производственного подразделения (Role Group), которое поддерживает осуществление функций, содержащихся в диаграмме, плюс одна нерабочая (нижняя) дорожка.
	+ Все объекты исходной IDEF3-диаграммы в процессе генерации Swim Lane автоматически размещаются на ее рабочих дорожках, если этим объектам в процессе разработки диаграммы были назначены роли. Если же назначение ролей не было сделано заранее, все объекты размещаются на нерабочей дорожке (не имеющей владельца). В этом случае распределение объектов исходной диаграммы между дорожками производится автором вручную. После окончания распределения объектов нерабочая дорожка остается пустой.
		1. Ход работы
1. Работа начинается с формирования словарей ролей производственного подразделения. Перейдите в словарь производственных подразделений (команда меню Dictionary → Role Group…) и внесите производственное подразделение «Отдел сборки компьютеров», для которого укажите уровень важности (Importance) как «средний» (Medium).
2. По аналогии создайте еще одно производственное подразделение – «Отдел продаж» с уровнем важности «средний».
3. Далее перейдите в словарь исполнителей (команда меню Dictionary → Role…) и внесите исполнителей согласно таблице 3.3.

Таблица 3.2 – Словарь исполнителей (Roles)

|  |  |
| --- | --- |
| Исполнитель | Производственноеподразделение |
| Диспетчер | Отдел сборки компьютеров |
| Менеджер по продажам | Отдел продаж |
| Сборщик ПК | Отдел сборки компьютеров |
| Специалист по программномуобеспечению | Отдел сборки компьютеров |
| Тестировщик | Отдел сборки компьютеров |

1. После того как будут созданы роли можно приступать к назначению исполнителей по каждой работе. Для этого в свойствах работы переходим на вкладку «Роли» (Roles) как показано на рисунке 3.6.



Рисунок 3.6 – Окно выбора роли работы «Подготовка комплектующих»

1. Укажите работам, объектам внешней ссылки и перекресткам роли согласно таблице 3.3.

Таблица 3.3 – Роли объектов диаграммы IDEF3

|  |  |
| --- | --- |
| Объект диаграммы | Роль |
| Подготовка комплектующих | Сборщик ПК |
| Установка материнской платы | Сборщик ПК |
| Установка процессора иоперативной памяти | Сборщик ПК |
| Установка SSD | Сборщик ПК |
| Установка HDD | Сборщик ПК |
| Установка видеокарты | Сборщик ПК |
| Установка FDD | Сборщик ПК |
| Установка DVD-ROM | Сборщик ПК |
| Установка Blu-Ray | Сборщик ПК |
| Инсталляция дополнительногопрограммного обеспечения | Специалист попрограммному обеспечению |

|  |  |
| --- | --- |
| Инсталляцияоперационной системы | Специалист попрограммному обеспечению |

1. Далее вызываем окно создания диаграммы дорожек (Swim Lane Diagram Wizard). Для этого выполняем команду меню Diagram → Add Swim Lane diagram…(см. рисунок 3.7)



Рисунок 3.7 – Окно Swim Lane Diagram Wizard – Step 1 of 2

1. Радиокнопкой выбираем базу для Swim Lane diagram из альтернативы Role Group или Text List UDP.
2. Выбрав Role Group из выпадающего списка, выбираем имя производственной группы (Role Group Name). На рисунке 3.7 выбран «Отдел сборки компьютеров».
3. Из выпадающего списка «Do you want to select a source Process Flow diagram» (Вы хотите выбрать в качестве источника диаграмму потоков работ?) выбираем нашу диаграмму IDEF3.
4. Формируем имя диаграммы (Diagram Name).
5. По кнопке «Далее» выходим на окно настроек «Swim Lane Diagram Wizard – Step 2 of 2» (рисунок 3.8).



Рисунок 3.8 – Окно Swim Lane Diagram Wizard – Step 2 of 2

1. Распределяем в этом окне роли на те, которые следует отобразить на диаграмме (Swim Lanes on Diagram) и на те, которые отображать не следует (Swim Lanes NOT on Diagram).
2. По окончанию настроек генерируем диаграмму дорожек, нажав кнопку «Готово».
3. В завершение производятся дизайнерский настройки диаграммы дорожек (Swim Lane diagram). Окончательный вид диаграммы представлен на рисунке 3.9.



Рисунок 3.9 – Swim Lane diagram

# 4 МЕТОДОЛОГИЯ DFD

* 1. Основные элементы методологии DFD

Методология DFD (Data Flow Diagram, Диаграмма потоков данных) являются основным средством моделирования функциональных требований проектируемой системы. С их помощью эти требования разбиваются на функциональные компоненты (процессы) и представляются в виде сети, связанной потоками данных. Главная цель таких средств продемонстрировать, как каждый процесс преобразует свои входные данные в выходные, а также выявить отношения между этими процессами.

Методология DFD поддерживает моделирование только информационных систем (подсистем), в то время как методология IDEF0 является основной для моделирования материально-информационных систем.

Модель DFD – иерархическая модель. Каждый процесс может быть подвергнут декомпозиции, то есть разбиению на структурные составляющие, отношения между которыми в той же нотации могут быть показаны на отдельной диаграмме.

Существуют два основных варианта: графические нотации Йордана-Де Марко (Yourdon and Coad Process Notation) и Гейна- Сарсона (Gane and Sarson Process Notation). Различие заключается в графических формах объектов. CA ERwin Process Modeler поддерживает нотацию Гейна-Сарсона**5**.

Основными компонентами диаграмм потоков данных являются элементы, представленные на рисунке 4.1.

5 Гейн К., Сарсон Т. Структурный системный анализ: средства и методы. Пер. с англ. Под ред. Козлинского А.В. М.: Эйтэкс, 1993



Рисунок 4.1 – Инструментарий методологии DFD (нотация Гейна-Сарсона)

К ним относятся:

* Внешние сущности (External Reference) представляют сущности вне контекста системы, являющиеся источниками или приемниками системных данных (материальный объект или физическое лицо). Одна внешняя сущность может одновременно предоставлять входы (функционирую как поставщик) и принимать выходы (функционирую как получатель). Определение некоторого объекта в качестве внешней сущности указывает на то, что он находится за пределами границ анализируемой системы, и такой объект не должен участвовать ни в какой обработке. Имя внешней сущности должно содержать существительное.
* Процессы (Activities) представляет собой преобразование входных потоков данных в выходные в соответствии с определенным алгоритмом, задаваемым именем процесса. Имя процесса должно содержать глагол в неопределенной форме с последующим дополнением.
* Накопители (хранилища) данных (Data Store) позволяют на определенных участках определять данные, которые будут сохраняться в памяти между процессами.

Фактически хранилище представляет «срезы» потоков данных во времени. Информация, которую они содержат, может использоваться в любое время после ее определения, при этом данные могут выбираться в любом порядке. Имя накопителя данных 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.

* Потоки данных (Precedence Arrow) являются механизмами, использующимися для моделирования передачи информации (или даже физических компонент) из одной части системы в другую. Потоки на диаграммах изображаются именованными стрелками, ориентация которых указывает направление движения информации.

При создании моделей DFD необходимо придерживаться следующих правил:

* Каждый процесс должен иметь хотя бы один вход и один выход. Процесс должен запускаться на выполнение либо через обрабатываемый, либо через управляющий поток данных. Работа каждого процесса должна завершаться конкретным результатом.
* Процесс обработки данных должен иметь внешнюю входящую стрелку (данные от внешней сущности). Для того, чтобы любой подобный процесс начал работать, мало использовать данные из хранилища, должна поступить новая информация для последующей обработки.
* Стрелки не могут связывать напрямую накопители данных, все связи идут через процессы. Нет смысла просто перемещать данные из одного места в другое.
* Каждый накопитель данных должен иметь как минимум один входящий и один выходящий поток. Наличие

только входящих потоков в накопитель означает, что информация накапливается, но не используется. В тоже время, прежде чем использовать данные из накопителя, они должны там появиться в результате работы какого- либо процесса (подсистемы, внешней сущности). Исключением из правил считается случай, когда накопитель является внешней сущностью. Тогда допускается наличие либо только входящих стрелок, либо только выходящих стрелок.

* 1. Лабораторная работа № 7. Построение диаграммы декомпозиции в нотации DFD
		1. Аннотация

Цель работы – получить навыки создания и редактирования моделей методологии DFD в CA ERwin Process Modeler.

При декомпозиции работы IDEF0 в DFD необходимо выполнить следующие действия:

* удалить граничные стрелки на диаграмме DFD;
* создать соответствующие внешние сущности и хранилища данных
* создать внутренние стрелки, начинающиеся с внешних сущностей вместо граничных стрелок;
* стрелки на диаграмме IDEF0 затуннелировать.

Строго придерживаться правил нотации DFD не всегда удобно, поэтому CA ERwin Process Modeler позволяет создавать в диаграммах потоков данных граничные стрелки.

В качестве примера в лабораторной работе рассматривается бизнес-модель деятельности компании, построенная в предыдущих лабораторных работах. В частности, декомпозиции подвергается функциональный блок А3 «Хранение и отгрузка».

В этой работе мы выделим следующие дочерние работы:

* снабжение необходимыми комплектующими – занимается действиями, связанными с поиском подходящих поставщиков и заказом у них необходимых комплектующих;
* хранение комплектующих и собранных компьютеров;
* оформление выдачи товара – занимается действиями, связанными с оформлением документации на отгрузку готовой продукции;
* отгрузка готовой продукции  все действия, связанные с упаковкой и собственно отгрузкой готовой продукции.
	+ 1. Ход работы
1. Выделим функциональный блок «Хранение и отгрузка» диаграммы деятельности А0 «Деятельность предприятия по сборке и продаже компьютеров и ноутбуков», нажмем на кнопку

 «Go to Child Diagram» (Перейти в дочернюю диаграмму) панели инструментов и выберем нотацию DFD.

1. При создании дочерней диаграммы CA ERwin Process Modeler переносит граничные стрелки родительского функционального блока, их необходимо удалить и/или заменить на внешние сущности. Стрелку «Бухгалтерская система» заменим на внешнюю сущность – кнопка  «External Reference Tool» (Инструмент «Внешняя сущность») на панели инструментов, в появившемся окне (рисунок 4.2) выбрать переключатель «Arrow» (Стрелка) и выбрать из списка нужное название. Остальные стрелки задействованы не будут, чтобы не загромождать диаграмму менее существенными деталями.



Рисунок 4.2 – Добавление внешней сущности

1. Добавим по аналогии и другие внешние сущности, которые будут необходимо для отображения поток данных в рамках рассматриваемой деятельности «Хранение и отгрузка»:
* персонал производственного отдела;
* система оформления заказов;
* поставщики;
* клиенты.
1. Далее разместим дочерние работы, свяжем их с внешними сущностями и между собой как представлено на рисунке 4.3.

Центральной здесь является работа «Хранение комплектующих и собранных компьютеров». На ее вход поступают собранные компьютеры, сопровождаемые их

перечнем, а также комплектующие от поставщиков, также сопровождаемые перечнем и сопутствующими документами на поставку. Выходом этой работы будут документы на поставку комплектующих, загружаемые в бухгалтерскую систему.



Рисунок 4.3 – Работы и внешние сущности

Другой задачей является «Снабжение необходимыми комплектующими». В рамках этой задачи мы получаем от персонала организации список необходимых комплектующих, передаем им их, и по необходимости осуществляет заказ не достающих комплектующих у поставщиков.

В рамках задачи «Оформление выдачи товара» мы обрабатываем заказы клиентов и готовим документы на поставку продукции, которые передаются на вход работы «Отгрузка готовой продукции». Затем в рамках этой задачи готовая продукция отгружается клиенту вместе с документами на поставку, полученными из бухгалтерской системы и делается отметка о доставке заказа клиенту в системе оформления заказов.

1. Следующим шагом необходимо определить, какая информация необходимо для выполнения каждой работы и учет какой информации необходимо вести, то есть необходимо разместить на диаграмме хранилища данных (рисунок 4.4).



Рисунок 4.4 – Итоговая диаграмма потоков данных

Работа «Хранение комплектующих и собранных компьютеров» работает с информацией о хранимых на складе комплектующих и собранных компьютерах. Стрелка, соединяющая работу и хранилища «Список комплектующих» и

«Список собранных компьютеров», двунаправленные, так как работа может получать информацию об имеющихся на складе комплектующих и компьютерах, так и вносить новые позиции.

Работа «Снабжение необходимыми комплектующими» работает с информацией о хранимых на складе комплектующих, а также с информацией о сделанных у поставщиков заказах. Стрелка, соединяющая работу и хранилище «Список комплектующих», однонаправленная, так как мы только получаем информацию о наличие комплектующих на складе. Стрелка, соединяющая работу и хранилище «Список заказов комплектующих», двунаправленная, так как работа может получать информацию об уже оформленных заказах, так и создавать и вносить новые заказы на комплектующие.

Наконец, работа «Оформление выдачи товара» работает со списком собранных компьютеров и вносит в него отметку о выдаче (передаче на отгрузку) проданных компьютеров, поэтому

стрелка между работой и хранилищем «Список собранных компьютеров» двунаправленная.

1. Последним действием необходимо стрелки родительской работы затунеллировать (рисунок 4.5).



Рисунок 4.5 – Диаграмма IDEF0 с затунеллированными стрелками работы «Хранение и отгрузка»

# 5 ЗАДАНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ

Пользуясь пакетом CA ERwin Process Modeler, разработать функциональную модель процесса предметной области, указанной в задании. Функциональная модель должна иметь не менее 3-х уровней декомпозиции (контекстная диаграмма А-0 в расчет не берется). Проверить синтаксис построенной модели. Сформировать отчет по модели IDEF0 в виде Node Tree.

Варианты заданий:

1. Разработать функциональную модель процесса учета товаров на складе.
2. Разработать функциональную модель процесса поиска книг в библиотечном каталоге.
3. Разработать функциональную модель процесса регистрации нового предприятия.
4. Разработать функциональную модель процесса получения кредита в банке.
5. Разработать функциональную модель процесса учета валютных операций в кассе банка.
6. Разработать функциональную модель процесса оптовой продажи товаров предприятия.
7. Разработать функциональную модель процесса формирования книг покупки и продаж предприятия.
8. Разработать функциональную модель процесса начисления зарплаты.
9. Разработать функциональную модель процесса платного приема больных в поликлинике.
10. Разработать функциональную модель процесса аккредитации образовательного учреждения.

# БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Mayer R.J. et al. Information integration for concurrent engineering (IICE): IDEF3 process description capture method report. Logistics research division, Wright-Patterson AFB, OH 45433
2. Гейн К., Сарсон Т. Структурный системный анализ: средства и методы. Пер. с англ. Под ред. Козлинского А.В. М.: Эйтэкс, 1993
3. Дубейковский В.И. Эффективное моделирование с CA ERwin Process Modeler (BPwin; AllFusion Process Modeler) – М.: Диалог-МИФИ, 2008 – 384 с.
4. Маклаков С.В. Моделирование бизнес-процессов с AIIFusion Process Modeler – М.: Диалог-МИФИ, 2008. – 224 с.
5. Р 50.1.028-2001. Информационные технологии поддержки жизненного цикла продукции. Методология функционального моделирования. М, 2002. – 54 с.
6. Репин В.В., Елиферов В.Г. Процессный подход к управлению. Моделирование бизнес-процессов. – М.: Манн, Иванов и Фербер, 2013. – 544 с.
7. Шемончук Д.С. Моделирование систем. Методические указания по выполнению лабораторных работ для студентов, обучающихся по направлению подготовки 230200 – М.: МИРЭА, 2009 – 32 с.