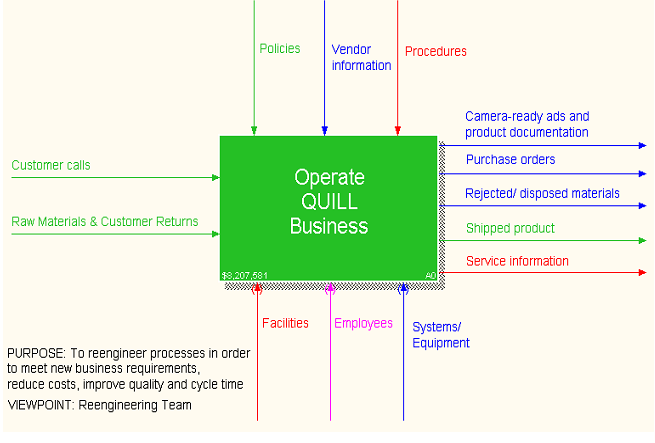


**V. Beșliu, P. Chirev, T. Scorohodova**

**Методические указания**

**к лабораторным работам**

**по дисциплине „Проектирование информационных систем”**



**Кишинёв**

**2016**

**Содержание**

**Введение.........................................................................................................................3**

**Лабораторная работа№ 1………………….………………………………………...5**

**Лабораторная работа№ 2 …………………………………………………….........14**

**Лабораторная работа№ 3 ………………………………………………………….25**

**Лабораторная работа№ 4 ………………………………………………………….31**

**Лабораторная работа№ 5 ……………………………………………………..…...35**

**Лабораторная работа№ 6 …………………………………………………...…......41**

**Лабораторная работа№ 7 …………………………………………………...…......45**

**Разработка курсового проекта ..……………………………………………..........48**

**Элементы управления проектами …………………………………………..……48**

**Ссылки ……………………………………………………………………….…........51**

**Методическое руководство по реализации лабораторных работ по курсу: Проектирование информационных систем (инструменты, руководства и методологические вопросы)**

**Бизнес-моделирование процессов на языках моделирования IDEF**

**Введение**

**IDEF** – сокращение от Integration Definition Metodology (Объединение Методологических Понятий). Семейство совместно используемых методов для процесса моделирования в сложных системах в различных разрезах и программной инженерии. Методы IDEF охватывают широкий спектр применения, от функционального моделирования данных, моделирования, объектно-ориентированного анализа / проектирования и приобретения знаний, т.е. направлены на различные методы описания и анализа процессов, потоков, структур промышленных и организационных систем с целью улучшения их характеристик. Методы взаимодополняют друг друга, обеспечивая возможность многоаспектного анализа систем. Методы обеспечивают научный подход к решению проблем. Методы разработаны с целью улучшения взаимодействия специалистов, занимающихся интеграцией существующих производственных и организационных систем.

Методы IDEF предлагают простые в использовании методы и стандартизированные языки представления информации. Также методы IDEF помогают пользователям:

* правильно понять текущую среду;
* предложить изменения;
* протестировать альтернативные решения;
* предсказать последствия изменений;
* успешно внести изменения.

Методы IDEF спроектированы для совместного применения как интегрированный концептуальный набор методов, которые могут быть связаны для описания развития целой системы. Каждый метод IDEF обращается к одному уникальному аспекту системы. Когда разные методы IDEF применяются вместе, они могут помочь достичь ключевые цели конкурентной инженерии, взяв во внимание различные факторы жизненного цикла системы в начале процесса проектирования. Достижение этой цели способно увеличивать преимущества интеграции, гибкости и реагирования компании.

В основе проектирования информационных систем лежит моделирование области знаний. Для создания проекта, соответствующего существующей области знаний (в форме приложения), необходимо обладать системным и комплексным представлением об области знаний, и разработать модели, которые отражают все аспекты функционирования будущей информационной системы. Под моделью области знаний понимается система, которая имитирует структуру или функционирование исследуемой области знаний и соответствует минимальному требованию – соответствие этой области знаний.

Для моделирования области знаний будут применены следующие нотации семейства IDEF:

* IDEF0 – функциональное моделирование (моделирование деятельности);
* IDEF1 – моделирование информации;
* IDEF1x – моделирование данных;
* IDEF3 – моделирование процессов, происходящих в системе (описания последовательности этапов процесса);
* IDEF4 - объектно-ориентированное проектирование.

Целью ставится построение функциональной схемы исследуемого объекта, которая описывает все процессы с достаточной точностью для однозначного моделирования информационной системы.

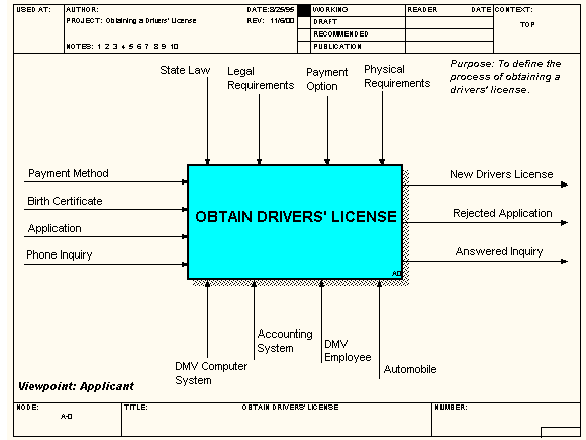


Рисунок 1 - Контекстная диаграмма

Построить функциональную схему исследуемого объекта можно построить при помощи методов IDEF.

**Функциональный методы в нотации IDEF:**

* IDEF0 – функциональное моделирование (моделирование деятельности);
* IDEF1 – моделирование информации;
* IDEF1x – моделирование данных;
* IDEF3 – описания последовательности этапов технологического процесса;
* IDEF4 - объектно-ориентированное проектирование;
* ABC – анализ бизнес затрат;
* функциональные методы потоков данных в нотации DFD.

Эти модели могут быть построены с помощью таких CASE-инструментов, как AllFusion Process Modeler (далее BPwin) и Erwin. Данные инструменты поддерживают нотации IDEF и DFD.

AllFusion Process Modeler 7 (BPwin) 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. В основу продукта заложены общепризнанные методологии моделирования, например, методология IDEF0 рекомендована к использованию Госстандартом РФ и является федеральным стандартом США. Простота и наглядность моделей Process Modeler упрощает взаимопонимание между всеми участниками процессов. Распространенность самого AllFusion Process Modeler 7 позволяет вести согласование функциональных моделей с партнерами в электронном виде. Продукт AllFusion Process Modeler 7 (BPwin) создан компанией Computer Associates. AllFusion Process Modeler 7 наряду с ERwin Data Modeler (ранее: ERwin), Data Model Validator (ранее: ERwin Examiner), Model Manager (ранее: ModelMart) входит в состав пакета программных средств AllFusion Modeling Suite, комплексное использование которого обеспечивает все аспекты моделирования информационных систем.

*Основные возможности системы*:

* поддержка различных технологий моделирования;
* анализ показателей затрат и производительности;
* интеграция процессов/данных;
* поддержка стандартных нотаций;
* экспорт объектов и свойств в другие модели;
* документирование информации в пределах всей модели;
* масштабируемость отчетности без потери качества графиков.

Источник: <http://pro-spo.ru/winmat/2663-allfusion-process-modeler>

**Лабораторная работа№ 1**

**Цель работы -** *Знакомство с нотацией IDEF0 и средой моделирования AllFusion Process Modeler (BPwin) [2]*

В данных рекомендациях будут применяться инструменты AllFusion Process Modeler (**BPwin**) <http://ca-allfusion-process-modeler.software.informer.com/7.2/>, который поддерживают нотации IDEF0, IDEF3, DFDи ABC и **Erwin,** который поддерживает нотацию IDEF1x. В процессе моделирования в среде BPwin можно перейти от нотации IDEF0 к нотации IDEF3 и DFD в любой ветви модели и тем самым построить комбинированную модель. Работа в среде AllFusion Process Modeler (**BPwin**) начинается с создания новой модели, необходимо указать название модели ***(Name***) и её тип ***(Type)*** (рисунок 1.1).

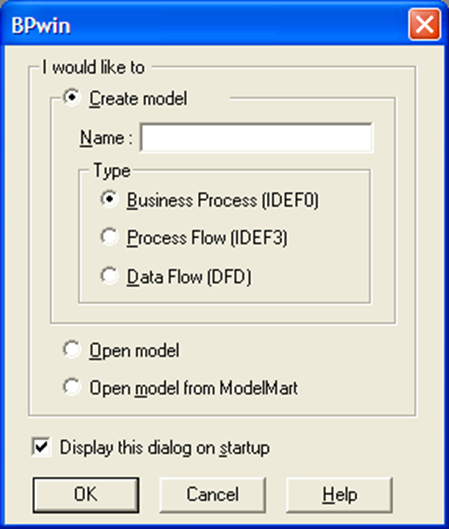
****

Рисунок 1.1 - Создание новой модели

Тип модели определяет вид декомпозиции объекта. При выборе типа модели *Business Process (IDEF0),* то в созданной модели возможно произвести декомпозицию процессов в нотациях IDEF0, IDEF3 или DFD; если выбрать тип модели *Data Flow (DFD)*, то в созданной модели композиция возможна в нотациях IDEF3 и DFD; тип модели *Process Flow (IDEF3)* позволяет выполнить композицию только в нотации IDEF3. После выбора названия модели и выбора типа модели необходимо нажать кнопку “OK”, и среда BPWin сразу же предложит задать параметры модели (рисунок 1.2).

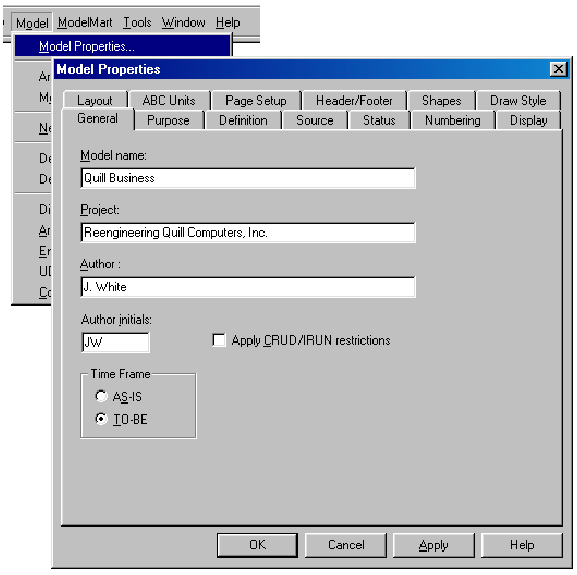


Рисунок 1.2 -Установка параметров модели

Где

*General —* автор модели;

*Numbering —* формат нумерации процессов и диаграмм, а также порядок отображения на диаграмме;

*Display —* список элементов, отображаемых на диаграмме;

*Layout —* параметры размещения;

*ABC Units —* единицы измерения для анализа бизнесс-затрат;

*Page Setup —* параметры затрат;

*Header/Footer* — параметры верхних и нижних колонтитулов.

После задания параметров модели необходимо нажать на кнопку “OK”, в результате чего появится диалоговое окно (рисунок 1.3), состоящее из пяти частей:

* (Model Explorer) – представление стуктуры модели (существующие диаграммы и их иерархия;
* (*Drawing Area)* – основная область окна, в которой отображаются диаграммы в процессе работы;
* панели инструментов: *Menu Bar*, *BPWin Toolbar, Standard Toolbar.*

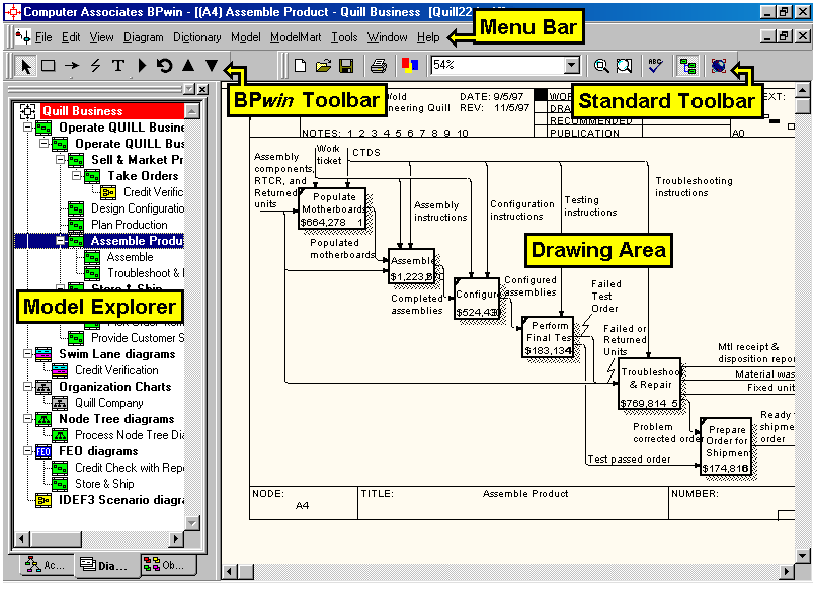


Рисунок 1.3 - Главная страница

**Панели инструментов *Model Toolbox***

**В понели инструментов** располагаются инструменты для создания различных графических элементов модели. Набор инструментов изменяется в зависимости от выбранной диаграммы. Тип и назначение кнопок на панели инструментов представлены в таблице 1.1.

Таблица 1.1 - Тип и назаначение кнопок на панели инструментов

|  |  |  |
| --- | --- | --- |
| **Кнопка** | **Название** | **Функция** |
|  | Pointer Tool | Преобразует курсор в стрелку для размещения элементов на диаграмме. |
| IEDF0  DFD  IDEF3 | Activity Box Tool | Размещение на диаграмме новых блоков. |
|  | Precedence Arrow Tool | Размещение на диаграмме новых стрелок для соединения блоков. |
|  | Squiggle Tool | Соединение стрелки с её названием. |
|  | Text Tool | Добавление текста на диаграмму. |
|  | Diagram Dictionary Editor | Активация окна управления диаграммами для визуализации существующих диаграмм и перехода выбранной диаграмме. |
|  | Go to Sibling Diagram | Переход между стандартной диаграммой, деревом узлов и диаграммой FEO. |
|  | Go to Parent Diagram | Переход к родительской диаграмме. |
|  | Go to Child Diagram | Переход к дочерней диаграмме. |
| -DFD | External Reference Tool | Размещение на диаграмме внешней сущности. |
| -DFD | Data store Tool | Размещение на диаграмме хранилища данных. |
| -IDEF3 | Junction Tool | Размещение на диаграмме перехода. |
| -DEF3 | Referent Tool | Размещение ссылки на диаграмме. |

Созданная диаграмма содержит контекстную диаграмму (рисунок 1.4) с одним функциональным блоком, которая была выбрана на этапе создания модели (в данном случае IDEF0). В последующем необходимо дать название этому функциональному блоку и по необходимости указать функциональные свойства.

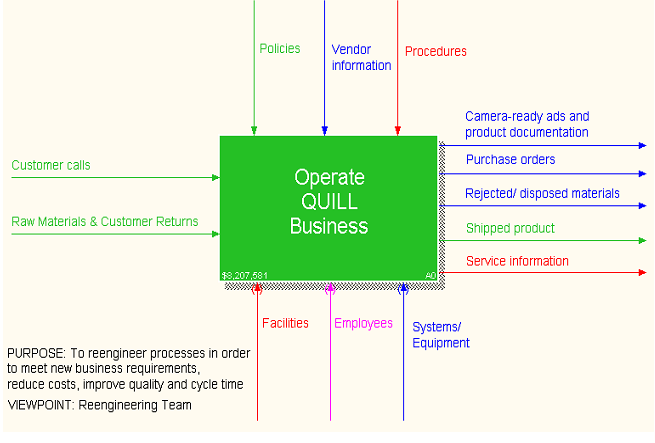


Рисунок 1.4 -Контекстная диаграмма

Для этого активируется диалоговое окно ***Activity Properties*** (рисунок 1.5)-двойной щелчок по левой кнопке мыши на функциональном блоке.

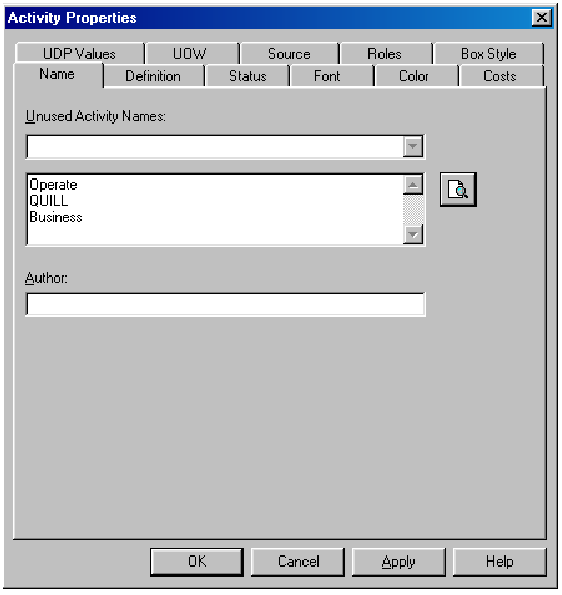


Рисунок 1.5- Диалоговое окно Activity Properties

Теперь на диаграмме можно добавить соединительные стрелки. Для этого требуется активировать иконку Precedence Arrow Tool на панели *Model Toolbox*, курсор становится крестиком. Далее необходимо выбрать точку, из которой будет начинаться соединительная линия, а затем точку, где она должна заканчиваться. (BPWin подсвечивает эти точки чёрным цветом при позиционировании курсора над ними). Для того, чтобы дать название этим аркам нажмите на иконку pointer tool (*Pointer Tool)* на панеле инструментов *Model Toolbox* и затем выполните двойной шелчок по желаемой стрелке, в результате появится диалоговое окно *Arrow Properties.* В поле “Arrow Name” задайте название соединения.

После того, как все входящие и исходящие стрелки размещены на диаграмме и их имена заданы, можно переходить к декомпозиции контекстной диаграммы на функциональные блоки. Для выполнения этого действия необходимо нажать на иконку  *Go to Child Diagram* на панели инструментов *Model Toolbox*. Теперь нужно выбрать работу, над которой следует провести декомпозицию. В результате появится диалоговое окно *Activity Box Count* (рисунок. 1.6), в котором выбирается нотация модели и количество функциональных блоков, на которые будет произведена декомпозиция (количество дочерних диаграмм).

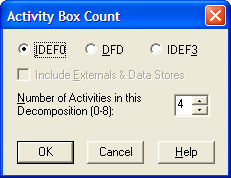


Рисунок 1.6 - Создание дочерних диаграмм

После создания дочерних диаграмм BPWin автоматически создаёт ранее заданное количество работ и размещает на странице “Master” (рисунок 1.7). Далее соедините стрелки с работами (функциональными блоками). В случае необходимости можно добавить на диаграмме стрелки, соединяющие данные функциональные блоки. Дальнейшая декомпозиция производится таким же способом.

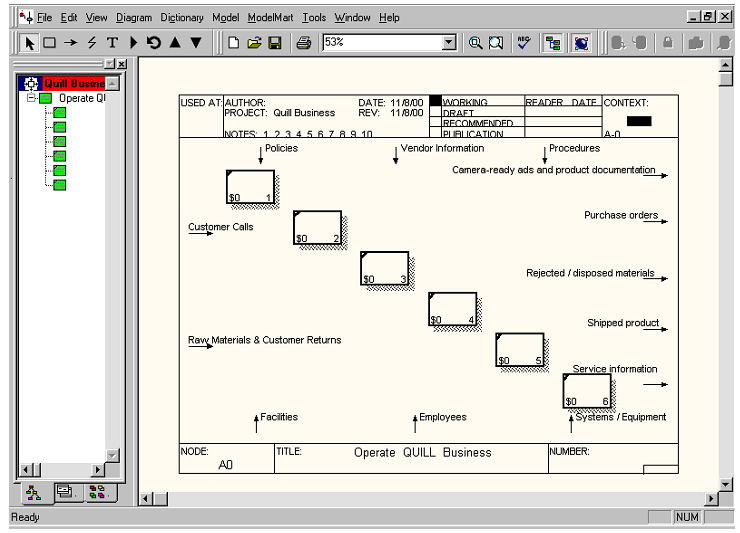


Рисунок 1.7 - Декомпозиция контекстной диаграммы

В некоторых случаях требуется соединить стрелку с множеством функциональных блоков одновременно. Для этого после соединения стрелки с одним функциональным блоком выбираем иконку Precedence Arrow Tool Arrow Tool на панели инструментов и выбираем необходимую работу (рисунки 1.8 – 1.12).

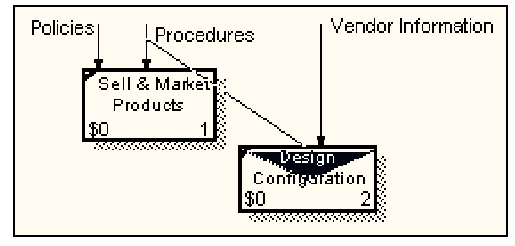


Рисунок 1.8 - Соединение стрелки с множеством работ

Для соединения стрелки с третьим функциональным боком следует щёлкнуть на желаемый сегмент (рисунок 1.9).

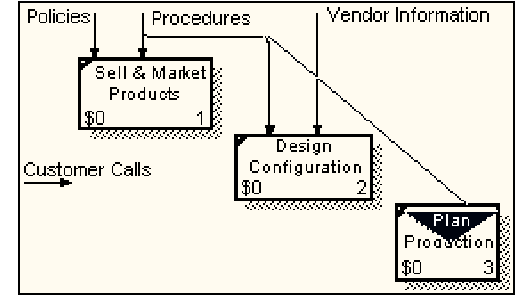


Рисунок 1.9 - Соединение стрелки с множеством работ

В ниже приведенном примере (рисунок 1.10) будет произведено соединение и с другими функциональными блоками.

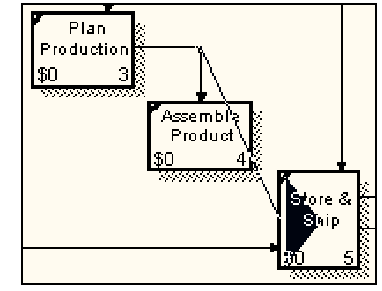


Рисунок 1.10 - Соединение стрелки с множеством работ

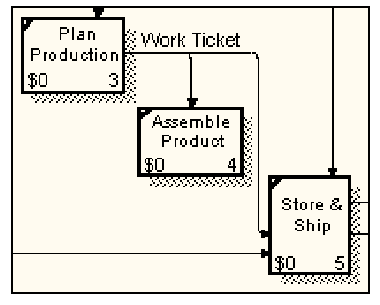


Рисунок 1.11 - Соединение стрелки с множеством работ

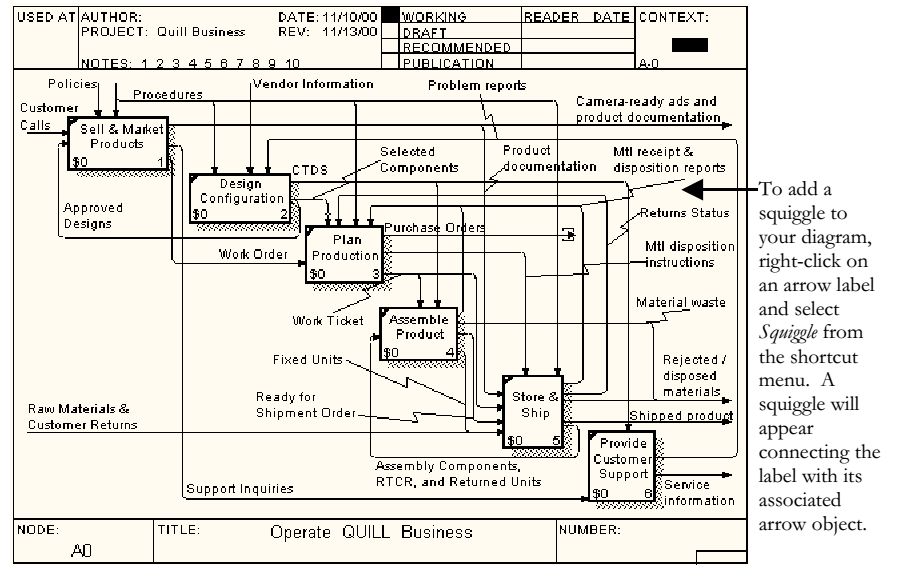


Рисунок 1.12 - Декомпозиция контекстной диаграммы

**Лабораторная работа№ 2**

*Целью этой работы является моделирование деятельности выбранной компании по стандарту IDEF0.*

**Задание № 1.** Разработка контекстной диаграммы в нотации IDEF0.

**Задание № 2.** Разработка диаграммы декомпозиции первого уровня в нотации IDEF0.

**Задание № 3.** Разработка диаграммы декомпозиции второго уровня в нотации IDEF0.

**Последовательноять моделирования бизнесс процессов компании**:

* выбор целевой области;
* описание целевой области (сфера деятельности, чем занимается объект, основные процессы в деятельности объекта);
* определение контекста моделирования;
* разработка контекстной диаграммы в нотации IDEF0;
* разработка диаграммы декомпозиции второго уровня в нотации IDEF0;
* разработка диаграммы декомпозиции следущего уровня в нотации IDEF0.

**Замечание**

*Вариант индивидуального задания согласуется с преподавателем. Все диаграммы в нотации IDEF0, IDEF3 и DFD будут построены в среде AllFusion Process Modeler.*

Цель этой лабораторной работы является моделирование деятельности выбранной компании, используя методологии:

* IDEF0 – методология функционального моделирования,
* IDEF3 – методология описания процессов,
* DFD – методология моделирования потоков данных,
* IDEF1X – методология моделирования данных.

Диаграммы первых трёх методологий будут быть построены в среде CASE- AllFusion Process Modeler BPWin и IDEF1X – в среде AllFusion ERwin Data Modeler.

Все диаграммы в нотациях IDEF0, IDEF3, DFD предназначениы для описания бизнесс процессов компании.

*Любая деятельность или набор деятельностей, в которых используются ресурсы, преобразующие входные данные в выходные, могут считаться процессами. Для результативного функционирования организации необходимо определить множество независимых и взаимодействующих процессов, а также управлять ими. Часто выход одного процесса определяет вход другого процесса.*

В результате моделирования бизнесс процессов компании будет получена модель бизнесс процессов, которая может быть трёх типов:

* модель AS-IS (как есть) – модель существующих бизнесс процессов организации;
* модель TO-BE (как будет) – модель идеальной организации бизнесс процессов предприятия;
* модель SHOULD-BE (как должно быть) – идеальная модель, но не отражающая реальную организацию бизнесс процессов предприятия.

На лабораторных работах будут построены модели AS-IS или TO-BE.

**Задание № 1. Разработка контекстной диаграммы в нотации IDEF0**

Перед тем, как перейти к построению диаграмм моделирования компании, рекомендуется изучить детали исследуемой области, для которой будет разработан проект информационной системы.

В приведённом примере моделируется компания по сборке компьютеров (ПК и ноутбуки), данная компания не занимается производством деталей или целых узлов, а выполняет сборку, тестирование собранных продуктов и их оптовую продажу.

Основные процедуры компании:

* сотрудники исследуют потребности рынка в ПК и ноутбуках;
* продавцы получают заказы от клиентов;
* сотрудники группируют заказы по типам компьюторов;
* сотрудники исследуют рынок поставщиков деталей и узлов;
* сотрудники отдела закупок заказывают и получают детали и узлы, необходимые для сборки компьютеров;
* сотрудники планируют производство на короткие и продолжительные сроки;
* сотрудники выполняют расчёт себестоимости и стоимости доставки готовой продукции;
* сотрудники собирают и тестируют компьютерную технику;
* сотрудники обеспечивают гарантийное обслуживание;
* сотрудники ведут учёт готовой продукции на складе, учёт продаж, учёт запасных частей;
* сотрудники упаковывают готовую продукцию согласно заказам, полученным от клиентов;
* сотрудники доставляют заказанную продукцию клиентам;
* сотрудники ведут бухгалтерский учёт, учёт труда, совершенствуют заказ, выдают чеки об оплате, получаю оплату согласно чекам.

Построение модели в нотации IDEF0 начинается с определения **контекста моделирования**, который включает ***субъект моделирования***, ***цель моделирования*** и ***точку зрения*** относительно модели.

Под **субъектом моделирования** понимается система сама по себе, в этом случае необходимо указать что входит в систему, что не является частью системы – ограничения системы и какие элементы будут считаться элементами внешней среды системы.

***Цель моделирования.*** Не возможно построить модель без определения цели. Цель должна отвечать на как минимум следующий список вопросов:

* Какова причина моделирования данного процесса?
* Что модель должна отображать?
* Какую информацию получить человек, изучающий эту модель?

***Точка зрения.*** Точказрения - это восприятие человека, который наблюдает систему в желаемом аспекте моделирования. Известен факт, что при изучении **субъекта моделирования** и при создании модели проходят подготовку специалисты в разных областях знаний (аналитики в рассматриваемой области, информатики, технические специалисты, экономисты), взгляды которых на процессы могут быть различны, однако модель должна быть построена в соответствии с одной точкой зрения. Точка зрения должна соответствовать цели моделирования и в процессе моделирования нельзя отходить от выбранной точки зрения.

В данной работе **субъект моделирования** – это “Организация”, а именно процессы, которые протекают внутри данной организации. **Цель моделирования** *-**это*бизнесс процессы, которые будут протекать в этой организации (модель TO-BE). ***Точка зрения –*** этовзгляд директора организации, как человека, знающего в общем всю структуру организации.

После определения **контекста моделирования** можно начать построение контекстной диаграммы (“чёрный ящик”). На ней отмечаются входы и выходы без детализации компонентов (функциональных блоков). Эта диаграмма состоит только из одного блока, представляющего целую организацию (рисунок 2.1 и рисунок 2.2.).



Рисунок 2.1 – Контекстная диаграмма организации

**Входы** – представляют информацию или материалы, которые будут обработаны данной работой (блоком).

**Выходы** – информация или материалы, которые являются результатом данной работы (блока).

**Регламенты (контроль)** – процедуры, правила, стратегии, стандарты и нормы, которыми руководствуется работа (блок).

**Механизмы** – ресурсы, которые обеспечивают выполнение работ (сотрудники, оборудование, базы данных и другое).

Для данной организации определены:

**Входные стрелки:**

* заказы клиентов – список компьютеров и ноутбуков, а также запрашиваемые клиентами комплектации;
* детали и узлы от поставщиков;
* информация (коммерческие предложения) от поставщиков;
* информация о рыночном спросе.

**Выходные стрелки:**

* готовая продукция – собранные компьютеры и ноутбуки;
* заказы поставщикам – список узлов, деталей и материалов, необходимый компании;
* оплата счетов поставщикам – оплата за узлы, детали и материалы, необходимые организации;
* маркетинговые материалы собственной продукции – прайс-лист, реклама.

**Управляющие стрелки** (регламентирование):

* *номативно-правовая база* – различные законодательные акты, регламентирующие деятельность организации;
* *правила и процедуры* – правила и процедуры, регламентирующие процессы производства (правила сборки, правила тестирования, нормы сборки продукции, внутренние стандарты для собранной продукции, процедура взаимоотношений с клиентами, нормы охраны труда).

**Стрелки механизмов:**

* человеческие ресурсы, (инженеры IT, программисты, маркетологи, экономисты);
* система бухгалтерского учёта;
* система учёта контрактов;
* техническое оборудование;
* транспортные единицы.



Рисунок 2.2 – Контекстная диаграмма компании

**Содержание отчёта:**

* название выбранного субъекта;
* описание исследуемой области (область деятельности, чем занимается объект, основные процессы, имеющие место в деятельности объекта);
* описание контекста;
* контекстная диаграмма.

**Задание № 2**

**Разработка диаграммы декомпозиции первого и второго уровня в нотации IDEF0**

**Цель работы:**

Построить диаграмму декомпозици первого уровня в нотации IDEF0.

В предыдущем задании была построена контекстная диаграмма, состоящая из единственного блока, который отражает деятельности организации в общем (на макро уровне) без детализации основных компонентов организации. В этой работе будут выделены основные подразделения организации, которые обеспечивают её функционирование, а затем построены диаграммы декомпозиции первого и второго уровня в нотации IDEF0.

**Декомпозиция** означает разделение сложного объекта на части (компоненты), которые взаимодействуют между собой.

В предыдущем задании были идентифицированы основные процедуры, которые обеспечивают процесс производства компьютеров и ноутбуков.

Основные процедуры компании:

* сотрудники исследуют потребности рынка в ПК и ноутбуках;
* продавцы получают заказы от клиентов;
* сотрудники группируют заказы по типам компьюторов;
* сотрудники исследуют рынок поставщиков деталей и узлов;
* сотрудники отдела закупок заказывают и получают детали и узлы, необходимые для сборки компьютеров;
* сотрудники планируют производство на короткие и продолжительные сроки;
* сотрудники выполняют расчёт себестоимости и стоимости доставки готовой продукции;
* сотрудники собирают и тестируют компьютерную технику;
* сотрудники обеспечивают гарантийное обслуживание;
* сотрудники ведут учёт готовой продукции на складе, учёт продаж, учёт запасных частей;
* сотрудники упаковывают готовую продукцию согласно заказам, полученным от клиентов;
* сотрудники доставляют заказанную продукцию клиентам;
* сотрудники ведут бухгалтерский учёт, учёт труда;
* сотрудники совершенствуют заказ, выдают чеки об оплате, получаю оплату согласно чекам.

Эти процедуры будут сгруппированы для выделения основных подразделений организации, обеспечивающих её функционирование.

**Управление**

* сотрудники ведут бухгалтерский учёт, учёт труда;
* сотрудники совершенствуют заказ, выдают чеки об оплате, получаю оплату согласно чекам.
* сотрудники ведут учёт готовой продукции на складе, учёт продаж, учёт запасных частей;
* сотрудники выполняют расчёт себестоимости и стоимости доставки готовой продукции;
* сотрудники планируют производство на короткие и продолжительные сроки.

**Маркетинг и продажи**

* сотрудники исследуют потребности рынка в ПК и ноутбуках;
* сотрудники исследуют рынок поставщиков деталей и узлов;
* продавцы получают заказы от клиентов.

**Производство (сборка и тестирование)**

* сотрудники группируют заказы по типам компьюторов;
* сотрудники собирают и тестируют компьютерную технику;
* сотрудники обеспечивают гарантийное обслуживание.

**Приобретение и поставка**

* сотрудники упаковывают готовую продукцию согласно заказам, полученным от клиентов;
* сотрудники доставляют заказанную продукцию клиентам;
* сотрудники отдела закупок заказывают и получают детали и узлы необходимые для сборки компьютеров.

Таким образом было выделено 4 основных подразделения организации, которые обеспечивают её функционирование. В таком виде и произведём первую декомпозицию:

* ***Управление;***
* ***Маркетинг и продажи;***
* ***Производство (сборка и тестирование);***
* ***Приобретение и поставка.***

Для выполнения декомпозиции необходимо активировать иконку  *Go to Child Diagram* и затем выбрать работу, над которой необходимо произвести декомпозицию. После этого отобразится окно *Activity Box Count* (рисунки 2.3 – 2.6), в котором необходимо выбрать нотацию модели и количество функциональных блоков, на которое будет произведена декомпозиция (количество дочерних диаграмм), выполняем те же действия, как и в лабораторной работе 1.

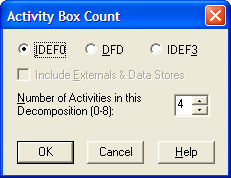


Рисунок 2.3 – Создание декомпозиции первого уровня



Рисунок 2.4 – Декомпозиция первого уровня

Далее даём название функциональным блокам и соединяем с ними краевые стрелки.



Рисунок 2.5 – Присвоение названий блокам и соединение с граничными стрелками

Далее необходимо соединить функциональные блоки для обеспечения функционирования процессов производства и получения готовой диаграммы первого уровня.



Рисунок 2.6 – Функциональные блоки. Диаграмма декомпозиции первого уровня

**Задание № 3**

**Разработка диаграммы декомпозиции второго уровня в нотации IDEF0**

На диаграмме декомпозиции первого уровня (рисунок 2.6) видно, что функциональных блоков ***Маркетинг и Продажи и*** ***Сборка и Тетстирование*** множество взаимосвязей. Это говорит о том, что в этих блоках проходит больше процессов, а значит необходимо произвести декомпозицию этих блоков до второго уровня (рисунок 2.7).



Рисунок 2.7 – Диаграмма декомпозиции второго уровня для блока ***Маркетинг и Продажи***

В процессе анализа процессов, протекающих при тестировании внешнего и внутреннего рынка были выделены 4 функциональных блока (рисунок 2.8).

Аналогично проводится декомпозиция для функционального блока “Сборка и Тестирование”.



Рисунок 2.8 – Диаграмма декомпозиции второго уровня для блока **Сборка и Тестирование**

При анализе прцессов протекающих при сборке продукции было выделено 4 функциональных блока (рисунок 2.9).



Рисунок 2.9 - Диаграмма декомпозиции второго уровня для блока **Сборка и Тестирование**

**Лабораторная работа№ 3**

**Построение диаграммы декомпозиции в нотации IDEF3**

**Цель работы:**

*Декомпозиция процессов, имеющих место в организации, в нотации IDEF3, также назывемая диаграмма потоков данных, — методология моделирования, использующая графическое описание потоков данных, взаимодействие процессов обработки информации и объектов, которые являются частью этих процессов* *[4].*

Диаграммы потоков данных будут использованы при моделировании бизнесс-процессов для представления процесса обработки информации. С помощью них будут описана деятельность сотрудников компании, например, последовательность обработки заказов. Построим диаграмму декомпозиции в нотации IDEF3 одной из работ в рамках лабораторной работы № 2.

Нотация IDEF3 позволяет аналитикам описать объекты в тех случаях, когда процессы происходят в определённой последовательности и взаимодействуют. Диаграммы в нотации IDEF3 могут содержать *“работы”, “связи”, “пересечения”* и *“объекты”*.

**Работы (Unit of Work, activity).** Представляются прямоуголниками (рисунок 3.1) и имеют название – отглагольное существительное, описывающее последовательность действий, может иметь идентификатор (номер); например, «производство продукта». В каждую сторону прямоугольника / из стороны прямоугольника может входить / выходить только одна стрелка.

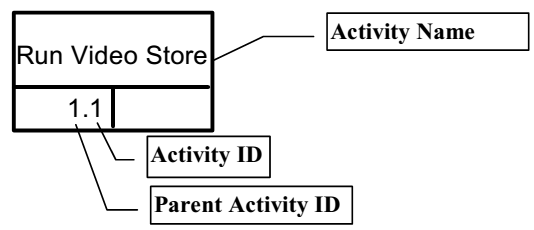


Рисунок 3.1 – Представление работы в IDEF3

**Связи.** Связи отображают отношения между работами. Все связи в нотации IDEF3 однонаправленные и могут быть направлены в любом направлении, но обычно диаграммы в нотации IEDF3 строятся таким образом, чтобы стрелки были направлены слева направо. Нужно отметить, что в нотации IDEF3 существуют три типа связей (стрелок) (таблица 3.1).

Таблица 3.1 - Описание стрелок

|  |  |  |
| --- | --- | --- |
| **Изображение стрелки** | **Название** | **Описание** |
| http://khpi-iip.mipk.kharkiv.edu/library/technpgm/labs/lab05.images/precedence.jpg | Связь предшествования (Precedence) | Непрерывная стрелка соединяет работы (UOW). Она рисуется слева направа или сверху вниз. Стрелка показывает, что исходная работа должна быть закончена до того, как начнётся следующая работа. |
| http://khpi-iip.mipk.kharkiv.edu/library/technpgm/labs/lab05.images/object_flow.jpg | Поток объектов (Object Flow) | Стрелка с двумя наконечниками, с одной стороны используется для отображения того, что объект используется как минимум в двух работах, например, когда объект создаётся в одной из работ и используется в другой. |
| http://khpi-iip.mipk.kharkiv.edu/library/technpgm/labs/lab05.images/relational_link.jpg | Связь отношения (Relational Link) | Стрелка с пунктирной линией, использующаяся для изображения связей между единицами работ (UOW) и между единицами работ и объектами ссылок. Эти соединения определяются аналитиком, создающим диаграмму. |

**Пересечение (Junction).** Завершение работы может стать сигналом для начала множества работ, а также существуют ситуации, когда для начала работы необходимо завершение множества работ. Пересечения используются для того, чтобы отобразить логику соединения стрелок (для их объединения или разветвления), а также для отображения множества событий, которые должны быть завершены до начала следующей работы. Таким образом, пересечения можно разделить на пересечения для объединения стрелок (Fan-in Junction) и пересечения для разветвления стрелок (Fan-out Junction). Стоит отметить, что пересечения не могут быть одновременно использованы для соединения и разветвления стрелок. Типы пересечений представлены в таблице 3.2.

Таблица 3.2 - Типы пересечений

|  |  |  |  |
| --- | --- | --- | --- |
| **Символ пересечения** | **Название** | **Смысл пересечения в случае объединения**  **(Fan-in Junction)** | **Смысл пересечения в случае разветвления (Fan-out Junction)** |
| http://khpi-iip.mipk.kharkiv.edu/library/technpgm/labs/lab05.images/a_and.jpg | *Асинхронный «&»*  AND | Все предшествующие процессы должны быть завершены. | Все последующие процессы должны быть запущены. |
| http://khpi-iip.mipk.kharkiv.edu/library/technpgm/labs/lab05.images/s_and.jpg | *Синхронный «&»*  (Synchronous AND) | Все предшествующие процессы должны быть завершены одновременно. | Все последующие процессы должны быть запущены одновременно. |
| http://khpi-iip.mipk.kharkiv.edu/library/technpgm/labs/lab05.images/a_or.jpg | *Асинхронный «sau»*  (Asynchronous OR) | Один или более предшествующих процессов должны быть завершены. | Один или более последующих процессов должны быть запущены. |
| http://khpi-iip.mipk.kharkiv.edu/library/technpgm/labs/lab05.images/s_or.jpg | *Синхронный «Sau»*  (Synchronous OR) | Один или более предшествующих процессов должны быть завершены одновременно. | Один или более последующих процессов должны быть запущены одновременно. |
| http://khpi-iip.mipk.kharkiv.edu/library/technpgm/labs/lab05.images/xor.jpg | *Исключающий (Исключающее или)*  XOR (Exclusive OR) | Только один предшествующий процесс должен быть завершён. | Только один из последующих процессов должен быть запущен. |

**Объект.** Объект в нотации IDEF3 представляет собой концепцию или данные, которые не могут быть соединены связями, пересечениями или работами. Объекты используются в моделе для того, чтобы читающий диаграмму обратил внимание на определённые важные аспекты модели. При создании объекта необходимо указать тип объекта (рисунок 3.2).

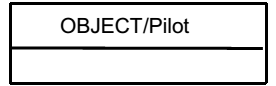


Рисунок 3.2 - Объект

В данной лабораторной работе будет выбрана работа с диаграммы IDEF0 и будет изучена детально благодаря методологии IDEF3. При декопозиции работ IDEF0 (и DFD) необходимо учитывать, что стрелки на данной диаграмме (IDEF0 и DFD) указывают потоки информации или потоки объектов, передаваемых от одной работы к другой.

На диаграммах IDEF3 стрелки показывают только последовательность выполнения работ, то есть они имеют отличное значение от стрелок на диаграммах IDEF0 или DFD. В процессе декомпозиции работ из IDEF0 или DFD в диаграммы IDEF3 стрелки не переносятся на диаграммы более низкого уровня. Если необходимо показать объекты с родительской диаграммы, то следует использовать ссылочные объекты.

**Изучим работу “Сборка и тестирование”, блок А3**

Выполнение этой работы начинается с размещения заказов на сборку. Первое действие начинается с:

* + проверка наличия достаточного количества узлов и деталей;
  + заказываются недостающие узлы и детали со склада.

Далее узлы и детали подготавливаются для сборки (извлечение из коробок, распаковка и т.д.). Затем начинается процесс сборки:

* + установка материнской платы в каркас;
  + установка центрального процессора;
  + установка оперативной памяти (RAM);
  + установка жёсткого диска.

Эти действия выполняются для каждого ПК и не зависят от конфигурации компьютера. На следующем этапе в зависимости от заказа клиента могут быть установлены:

DVD, TV- тюнер, Card-reader.

На этом заканчивается сборка компьютера.

Следующие операции:

* + установка операционной системы;
  + по требованию может быть установлено и другое программное обеспечение.

Наконец формируются отчёты: **”Отчёт по сборке ПК*”*** и **”Отчёт о результатах сборки”*.***

**Моделирование процесса “Сборка и тестирование” в нотации IDEF3**

Для проведения декомпозиции выделяем на диаграмме A3 работу **“ Сборка и тестирование”**,нажимаем иконку"Go to Child Diagram" и выбираем нотацию IDEF3.



Рисунок 3.3 – Исходная диаграмма первого уровня в нотации IDEF3

На дочерней диаграмме декомпозиции могут быть добавлены новые работы, поэтому при выборе количества работ можно указать произвольное количество работ (указываем количество блоков равным 4). (рисунок 3.3).

При создании дочерней диаграммы приложение BPWin не переносит граничные стрелки с родительской диаграммы, они должны быть заменены на ссылочные объекты:

***”размещение заказов”, ”узлы и детали”, ”заказ узлов и деталей”* ”*Отчёт о сборке ПК”*** и ***”Отчёт о результатах сборки”.***

Для этого необходимо нажать на иконку “R” (Referent) на панеле инструментов и в диалоговом окне (рисунок 3.4) выбрать пункт *“Arrow”*, а также из названий стрелок – необходимое соединение.

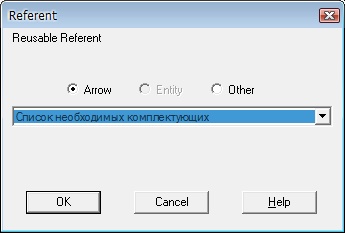


Рисунок 3.4 – Диалоговое окно Referent

Далее устанавливаем необходимые работы и соединяем их между собой в порядке протекания процесса сборки, в результате чего получается готовая диаграмма (рисунок 3.5).



Рисунок 3.5 – Диаграмма декомпозиции первого уровня в нотации IDEF3

Рассмотрим основные части этой диаграммы.

После выполнения работы **“*проверка наличия узлов и деталей”*** возможны два действия - заказ недостающих элементов со склада или, в случае наличия таковых, выполнить работу **“*подготовка узлов и деталей”***, по этой причине добавляем конъюнкцию http://khpi-iip.mipk.kharkiv.edu/library/technpgm/labs/lab05.images/xor.jpg “*Синхронный «ИЛИ»* (Synchronous OR)”. Работы **“*подготовка узлов и деталей”*** и **“*установка материнской платы и центрального процессора”*** соединяются со стрелкой **“*Потоки данных”*** http://khpi-iip.mipk.kharkiv.edu/library/technpgm/labs/lab05.images/object_flow.jpg. Это говорит о том, что между работами передаются объекты. Последующие работы соединяются стрелкой http://khpi-iip.mipk.kharkiv.edu/library/technpgm/labs/lab05.images/precedence.jpg ”***предшедствующее соединение (Precedence)”***, так как они показывают последовательность действий над теми же объектами. После установки жёсткого диска возможна установка DVD, ТВ-тюнера, Card-reader или иных компонентов. С этой целью используется соединение http://khpi-iip.mipk.kharkiv.edu/library/technpgm/labs/lab05.images/a_or.jpg “Асинхронный *«ИЛИ » (Asynchronous OR)”*. Соединения того же типа установливаются после окончания работ. После установке операционной системы по запросу клиента может быть установлено и другое программное обеспечение, или происходит заполнение отчётов. Для этого используется соедиенение http://khpi-iip.mipk.kharkiv.edu/library/technpgm/labs/lab05.images/xor.jpg *Исключение (исключённое «ИЛИ»)* XOR (Exclusive OR), так как после соединения http://khpi-iip.mipk.kharkiv.edu/library/technpgm/labs/lab05.images/xor.jpg может следовать только соединение http://khpi-iip.mipk.kharkiv.edu/library/technpgm/labs/lab05.images/xor.jpg , то перед работой “Заполнение отчётов” устанавливается соединение того же типа.

**Содержание отчёта:**

* Детальное описание работы, которую необходимо декомпозировать.
* Диаграмма декомпозиции.

**Лабораторная работа№ 4**

**Построение диаграммы декомпозиции в нотации DFD**

**Цель работы:**

*Определение потоков данных и построение диаграммы декомпозиции потоков данных в нотации DFD для одной из работ из предыдущей лабораторной работы****.***

Диаграммы потоков данных (Data flow diagram, DFD) используются для описания обращения документов и обработки информации. DFD, как и IDEF0, преставляет моделируемую систему в виде сети взаимосвязанных работ. Эти диаграммы могут служить дополнением диаграммам в нотации IDEF0 с целью представить более ясно текущие процессы циркуляции документов в корпоративной системе обработки информации. Основным назначением данных диаграмм является представление того, как каждая из работ преобразует входную информацию в выходную, а также связей между этими работами. Диаграмма в нотации DFD может содержать***: работы, внешние сущности, связи (потоки данных) и хранилища.***

***Работы*.** Представляются в виде прямоугольников с закруглёнными углами (рисунок 4.1). Смысл работ соответствует смыслу работ в нотациях IDEF0 и IDEF3. Работы, как и в нотациях IDEF3 и DFD, имеют входы и выходы, но не поддерживают управляющих связей и механизмов IDEF0. Все стороны блоков работ равноправны. В любую работу и из неё могут входить или выходить одна и более связей.

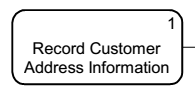


Рисунок 4.1 - Представление ***Работы***

***Внешние сущности*.** Внешние сущности отражают входы и выходы системы. Одна и та же ***внешняя сущность*** может одновременно иметь, как входы (функционируя в роли приёмника), так и выходы (имея роль поставщика). Внешние сущности – это материальные объекты, например, получатель, клиент, персонал, поставщик, склад. ***Внешние сущности*** находятся вне анализируемой системы. ***Внешняя сущность*** представляется прямоугольником с тенями на левой и верхней сторонах (рисунок 4.2).



Рисунок 4.2 – ***Внешние сущности***

***Соединения (потоки данных).*** Соединения представляют переход объектов от одной работы к другой, это значит, что диаграммы в нотации DFD не обладают краевыми стрелками. Соединения могут быть двунаправленными (рисунок 4.3).

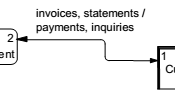


Рисунок 4.3 – Двунаправленное соединение

***Хранилища.*** Хранилище– это устройство для хранения информации, который позволяет записывать и считывать информацию в любой момент, кроме того методы записи и чтения могут быть различны(рисунок 4.4.). На самом деле хранилище является моделью будущей базы данных, то есть информация, сохранённая в хранилище, должна соответствовать требованиям модели(Entity-Relationship Diagram).



Рисунок 4.4 – *Хранилище* в DFD

**Декомпозиция работ IDEF0 в диаграммы DFD**

Для выполнения декомпозиции из нотации IDEF0 в нотацию DFD необходимо выполнить следующие действия:

* Удалить все краевые стрелки на диаграмме DFD.
* Построить внешние сущности и хранилища данных вместо краевых стрелок.
* Построить внутренние стрелки, начинающиеся с внешних сущностей, которые заменяют краевые стрелки.
* Туннелировать стрелки диаграммы в нотации IDEF0.

Необходимо отметить, что сложно строго соответствовать требованиям нотации DFD, и по этой причине, BPWin позволяет строить краевые стрелки на диаграммах DFD.

**Структура диаграммы декомпозиции в нотации DFD**

Выполним декомпозицию работы **“*Закупки и поставки”*** с диаграммы декомпозиции A0 – смотрите рисунок 1.9 в лабораторной работа 2. В этой работе A4 будут выделены работы дочерней диаграммы, а именно:

* Приобретение узлов и деталей – осуществление действий, связанных с выбором наиболее подходящих поставщиков, выполнение запроса по поставке компонентов.
* Хранение узлов, деталей и собранных компьютеров.
* Поставка готовой продукции – все действия по поставке: упаковка, заполнение документов и поставка продукции клиентам.

Выбираем работу **“*Закупки и поставка”*** на диаграмме A0, смотри рисунок 2.6. Лабораторная работа 2, нажимаем на иконку  "Go to Child Diagram" на панели инструментов и выбираем нотацию DFD.



Рисунок 4.5 – Исходная дочерняя диаграмма в нотации DFD

При создании дочерней диаграммы в приложении BPWin преобразуем краевые стрелки с родительской диаграммы (рисунок 4.5), как отмечено ранее, их необходимо удалить и заменить на внешние сущности – кнопка "External Reference Tool" на панеле инструментов, в появившемся окне выбираем переключатель "Arrow", а также нужное название из списка (рисунок 4.6).

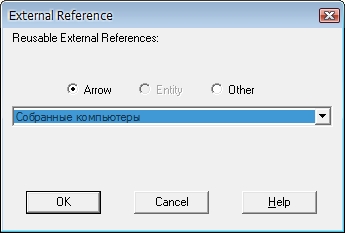


Рисунок 4.6 – Диалоговое окно “External Reference”

Располагаем дочерние работы и соединяем их с внешними сущностями (рисунок 4.7).



Рисунок 4.7 – Дочерняя диаграмма в нотации DFD

**Лабораторная работа№ 5**

**Цель работы:**

*Использование программного приложения AllFusion ERwin Data Modeler для проектирования базы данных в стандарте IDEF1X.*

*CA ERwin Data Modeler (далее ERwin) –* это CASE - инструмент для проектирования баз данных. Он позволяет создавать, документировать и выполнять мониторинг баз данных, хранилища и галереи данных [3].

При начале работы с приложением необходимо создать новую модель, выбрать её тип (Logical, Physical или Logical/Physical) и СУБД, которую планируется использовать, а также её версию (рисунок 5.1).

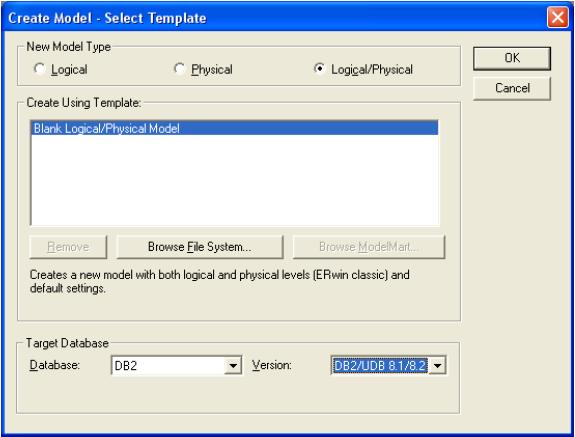


Рисунок 5.1 – Создание новой модели

ERwin позволяет создать модель логического уровня, модель физического уровня или комбинированную модель (Logical, Physical или Logical/Physical).

**Модель логического уровня** – это абстрактный подход к данным (информации). В данной модели данные представляются в форме, соответствущей реальности и могут называться в соответствии с их названиями в предметной области (“Поставщик”, “Клиент”, “Отдел”, “Заказ”).

Объекты на данной модели называются “сущностями” и “атрибутами”. Логическая модель данных универсальна и не связана с применением реальной СУБД.

**Модель физического уровня** – зависит от выбранной СУБД**.** В физической модели содержится информация о всех объектах базы данных. Физическая модель зависит от СУБД, в которой планируется создание базы данных.

Erwin на логическом уровне поддерживает нотации IE и IDEF1X, а на физическом ­– IE, IDEF1X и DM. Далее будет использована нотация IDEF1X.

Переход от логической модели к физической модели происходит путём выбора типа модели в выпадающем списке на панели инструментов (рисунок 5.2).

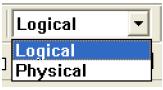


Рисунок 5.2 – Переход от логической к физической модели

#### **Логический уровень модели данных**

Для построения логической модели используется панель инструментов *Toolbox*. Используя инструменты, представленные на этой панели, можно добавить на диаграмму сущности и связи между ними (рисунок 5.3).

http://khpi-iip.mipk.kharkiv.edu/library/technpgm/labs/Lab08_Images/08_03.jpg

Рисунок 5.3 – Панель инструментов Toolbox

Ниже в таблице 5.1 даётся описание назначения иконок, представленных на рисунке 5.3.

Таблица 5.1 – Описание назначения инструментов Toolbox

|  |  |
| --- | --- |
| **Иконка** | **Назначение** |
| http://khpi-iip.mipk.kharkiv.edu/library/technpgm/labs/Lab08_Images/08_03_01.jpg | Создание новой сущности. Для создания новой сущности необходимо нажать на иконку и затем на свободное место на диаграмме. |
| http://khpi-iip.mipk.kharkiv.edu/library/technpgm/labs/Lab08_Images/08_03_02.jpg | Создание категорий. Для добавления категоризированного соединения необходимо нажать на иконку, выбрать родительскую сущность, а затем дочернюю сущность. |
| http://khpi-iip.mipk.kharkiv.edu/library/technpgm/labs/Lab08_Images/08_03_03.jpg | Добавление идентифицирующей связи. Для соединения двух сущностей идентифицирующей связью необходимо нажать на иконку, выбрать родительскую сущность и затем дочернюю сущность. |
| http://khpi-iip.mipk.kharkiv.edu/library/technpgm/labs/Lab08_Images/08_03_04.jpg | Добавление связи “многие ко многим”. |
| http://khpi-iip.mipk.kharkiv.edu/library/technpgm/labs/Lab08_Images/08_03_05.jpg | Добавление неидентифицирующей связи. |

После создания сущностей можно добавить к ним атрибуты. Существует два способа добавления атрибута: двойной клик по сущности на диаграмме; выбор в контекстном меню пункта *Attributes* (рисунок 5.4).

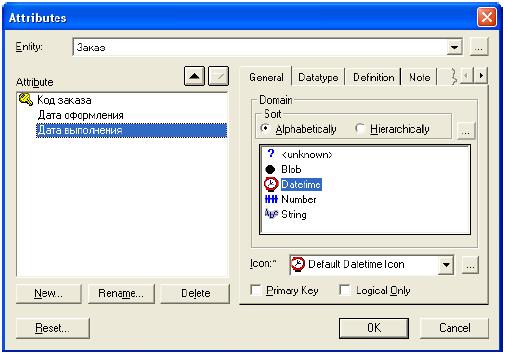


Рисунок 5.4 – Окно атрибутов выбранной сущности

В окне атрибутов сущности можно просмотреть и отредактировать информацию о созданных атрибутах, а также добавить новый атрибут. Есть возможность выбрать первичных ключ. Для добавления нового атрибута необходимо нажать кнопку “*New”*. В появившемся диалоговом окне (рисунок 5.5) можно выбрать тип атрибута (BLOB, дата/время, число, строка), задать имя атрибуту (*Attribute Name*) и название колонки (Column Name), которая будет представлена на физическом уровне (рисунок 5.5).

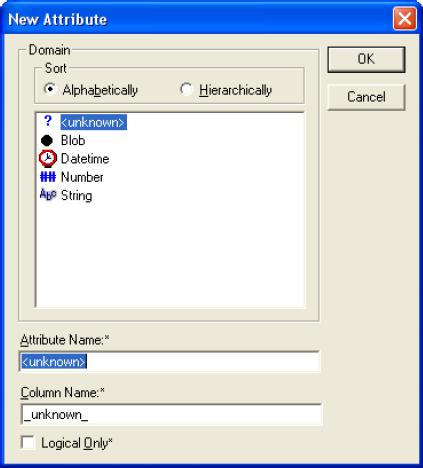


Рисунок 5.5 – Создание атрибута

После создания сущностей добавляем соединения между ними. При создании идентификационной связи атрибуты первичного ключа родительской сущности добавляются в состав первичного ключа дочерней сущности, а при создании неидентификационной связи атрибуты просто добавляются к атрибутам дочерней сущности.

Для определения или изменения типа связи необходимо осуществить двойной щелчок мыши на связи или выбрать из контекстного меню пункт *Relationship Properties* (рисунок 5.6). Здесь из доступных опций при выборе опции *General* можно присвоить название соединению (в направлении дочерняя сущность – родительская сущность, а также в направление дочерняя сущность – родительская сущность), мощность связи (ноль, идин и или более; один и более (Р); ноль или один (Z); фиксированная (определённое число), можно изменить тип связи. Выбрав опцию *RI Action* можно задать ограничения целостности.

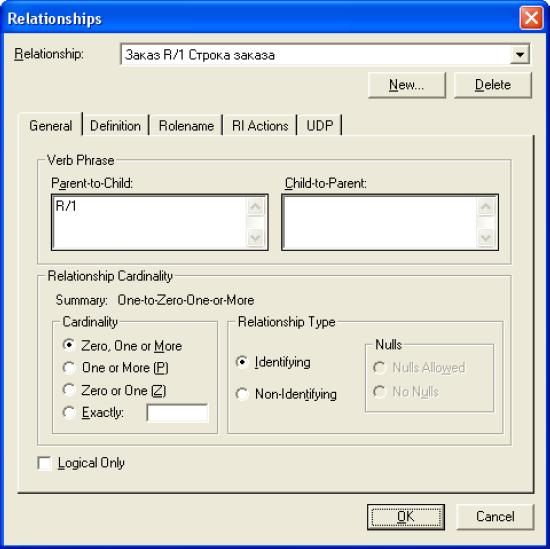


Рисунок 5. 6 - Окно “Relationship”

Пример логической модели базы данных представлен на рисунке 5.7.



Рисунок 5. 7 – Пример логической схемы базы данных

#### **Физический уровень модели данных**

При переходе с логического уровня на физический автоматически создаётся физическая схема базы данных (рисунок 5.8).

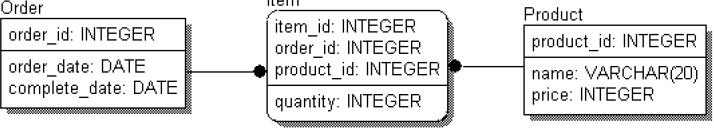


Рисунок 5.8 – Автоматически созданная физическая схема базы данных

Эта модель может быть дополнена или изменена. Принцип создания физической схемы аналогичен созданию логической схеме.

После создания физической схемы базы данных можно сгенерировать скрипты для заданной СУБД. С этой целью используется пункт меню *Tools* -> *Forward Engineering/Schema Generation* (рисунок 5.9).

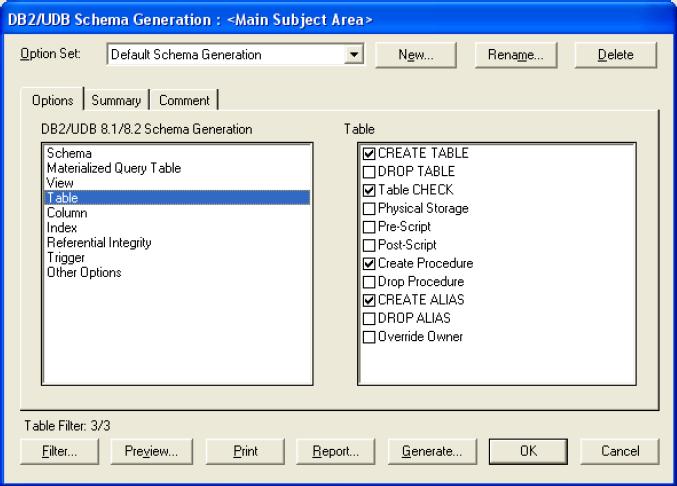


Рисунок 5.9 -Окно генерации SQL-скриптов для заданной СУБД

Здесь можно указать какие скрипты должны быть сгенерированы, предварительно их просмотреть и сгенерировать (Erwin выполнить подключение к выбранной СУБД и создаст SQLскрипт в автоматическом режиме).

Данная лабораторная работа может быть принята только вместе с лабораторной работой № 6 - "Разработка логической модели предметной области".

**Лабораторная работа № 6**

**“Разработка логической модели предметной области”**

**Цель работы:** *Разработка логической модели базы данных для выбранной области знаний в нотации IDEF1X.*

В этой лабораторной работе необходимо разработать логическую модель области знаний в нотации IDEF1X, используя CASE - инструмент ERwin Data Modeler. Логическая схема данных области знаний должна основываться на бизнесс процессах, которые были смоделированы в предыдущих лабораторных работах.

**IDEF1X**

IDEF1X основывается на методе Чена и позволяет моделировать данные реляционной базы данных в третьей нормальной форме.

***Сущность (Entity)*** – это реальный или вымышленный объект, который имеет определённое значение для данной области знаний. Каждая сущность должна иметь имя в виде существительного в единственном числе, а также обладать уникальным идентификатором. Сущности должны быть однозначно идентифиуцированы и должны отличаться от других сущностей данного типа.

***Атрибут (Attribute)*** – это любая характеристика **сущности***,* имеет значение для исследуемой области знаний и служит для классификации, идентификации, представлять качественную характеристику или отображать состояние сущности. Название атрибута должно быть существительным в единственном числе.

***Связи (Relationship)***– это ассоциация между двумя сущностями исследуемой области знаний.

В нотации IDEF1X ***сущности*** делятся на три категории: ***зависимые,*** ***независимые от идентификаторов*** или просто ***независимые,*** есликаждый экземпляр может быть однозначно идентифицирован без определения связей с другими ***сущностями.***

*Независимая сущность* графически может быть представленапрямоугольником, в то время, как *зависимая сущность* представляется прямоугольником с закруглёнными углами.

В нотации IDEF1X существуют следующие мощности связей:

* мощность N – каждый экземпляр родительской сущности может иметь 0 (ноль), один или более связанных экземпляров, связанных с одним экземпляром сущности-потомка;
* мощность Р – каждый экземпляр сущности-родителя должен иметь не менее одного экземпляра сущности-потомка, связанного с одним экземпляром сущности-родителя;
* мощность Z – каждый экземпляр сущности-родителя должна иметь не более, чем один экземпляр сущности-потомка, связанный с сущностью-родителем;
* фиксированное число – каждый экземпляр сущности-родителя должен быть связан с определённым числом сущностей-потомков*.*

В ERwin при создании ***идентифицирующей связи*** атрибуты первичных ключей родительской сущности автоматически переходят в состав первичного ключа сущности-потомка*.* Это называется миграцией атрибутов. В сущности-потомке эти атрибуты идентифицируются как внешний ключ (FK). При создании ***неидентифицирующего соединения*** атрибуты первичного ключа сущности-родителя автоматически переносятся в неключевые атрибуты сущности-потомка*.*

**Разработка логической модели данных организации по сборке и продаже ПК и ноутбуков**

Для этой организации будет разработана логическая модель процесса сборки компьютеров.

Для создания логической модели данных необходимо выполнить следующие шаги:

1. **Идентификация** бизнес требований.
2. **Анализ** бизнес требований.
3. Создание **концептуальной модели** данных. Одобрение модели представителями организации.
4. Создание **логической модели данных**, которая содержит:
   1. Выбор целевой СУБД (генерация скриптов для физической схемы).
   2. Создание **документа со стандартными абревиатурами** для логических и физических объектов.
   3. Определение **областей знаний.**
   4. Определение **правил**.
   5. Определение **значений по-умолчанию.**
   6. Создание **сущностей** и добавление определений.
   7. Установка **типов данных** атрибутам.
   8. Добавление **ограничений** (CHECK) или **значений по умолчанию**.
   9. Определение **первичных** и **уникальных** ключей.
   10. Определение **индексов.**
   11. Создание подтипов и **супертипов,** если **необходимо** (наследование).
   12. Определение **связей** между сущностями и создание **внешних ключей**.
   13. **Проверка** модели данных.
   14. **Утверждение** логической модели.

**Создание сущностей и добавление определений**

Для создания логической модели необходимо определить сущности исследуемой области. Для исследуемой области знаний идентифицируем следующие сущности:

* ***клиент*** – человек, покупающий компьютер;
* ***заказ*** – список компьютеров, заказанных клиентом;
* ***компьютер***;
* ***детали*** – элементы, из которых собираются компьютеры;
* ***сотрудник*** – специалист компании, который собирает конкретный компьютер,

Определим связи между этими сущностями:

* ***клиент – заказ***. Один клиент может разместить несколько *заказов*. Необходимо отметить, что если клиент занесён в базу данных, то значит он сделал хотя бы один заказ. Исходя из этого мощность *связи* *– Р*, связь является идентифицирующей, так как *заказ* не может существовать без *клиента*;
* ***заказ – компьютер***. *Клиент* может заказать несколько *компьютеров*, но как минимум один компьютер. Исходя из этого мощность *связи* *– Р*, связь является идентифицирующей, так как *компьютер* не может существовать без *заказа*;
* ***компьютер*** **–** **детали**. *Компьютер* может содержать множество различных *компонентов*, в то же время один и тот же тип компонентов может являться составной частью различных *компьютеров*. Мощность связи – *многие ко многим*. Данный тип связи не существует в IDEF1X. Чтобы решить эту проблему необходимо ввести понятие ассоциативная сущность – *конфигурация*;
* мощность связи между сущностями ***компьютер*** и ***конфигурация*** – **P**. Так как у любого *компьютера* обязательно должна быть *конфигурация*, то мощность связи между сущностями **детали** и **конфигурация** – **N**. Может случится такое, что новый компонент не был ещё установлен ни на один компьютер. Связь в обоих случаях является идентифиуцирующей, так как конфигурация компьютера не может существовать без прямой связи с *компьютером* и *деталями*;
* ***детали*** **– *тип деталей***. Очевидно то, что количество типов деталей является ограниченным. Мощность связи - **Р**. Связь является идентифицирующей;
* ***компьютер*** **–** **сотрудник**. Каждый компьютер собирается одним специалистом. Один и тот же сотрудник может собрать несколько компьютеров*.* Мощность связи – **N**. Связь является ***неидентифицирующей***, так как экземпляр сущности компьютер уже может существовать, но не быть связанным ни с одним специалистом. Именно поэтому в свойствах этой связи устанавливается свойство "Nulls Allowed" (представляется ромбом со стороны родительской сущности сотрудник).



Рисунок 6.1 – Логическая модель данных организации по сборке компьютеров и ноутбуков

Физический уровень модели зависит от выбранной СУБД. ERwin поддерживает более 20 типов СУБД (реляционные и не только):DB2 LUW 9.x; DB2 ZOS 9.x; Informix 11.x; Oracle 11g; MS SQL Server 2000,2005,2008,2012; MySQL 5.x; Sybase (ASE) 15.0; Sybase IQ 12.x; Teradata v13; ODBC 3.x; DB2 iSeries 5.x/6.x; SAS; Progress 10.x; SQL Azure и другие.

**Лабораторная работа № 7**

**Цель работы:**

*Разработка технической спецификации, необходимой для реализации информационной системы выбранной организации.*

Техническое задание ­– это документ, который описывает цели и задачи, требования и основные входные данные, необходимые для разработки информационной системы (Технический регламент 38370656- 002:2006).

**При разработке технического задания должны быть решены следующие вопросы:**

* установка цели создания информационной системы, определение функций и подсистем;
* разработка и обоснование требований к подсистемам;
* разработка и обоснование требований к информационной базе, математическим ресурсам, программам и технике (включая средства передачи данных);
* определение основных требований проектируемой системы;
* определение списка работ по созданию системы и исполнителей;
* определение этапов создания системы и сроков их выполнения;
* предварительный расчёт затрат на создание системы и определение уровня экономической эффективности её реализации.

**Техническое задание должно содержать следующие главы**:

* 1. **Общие положения**
* полное название системы и абревиатура;
* код (номер) контракта;
* название организации – исполнителя и бенефициара, их реквизиты;
* список документов, на основе которых создаётся система;
* дата начала и завершения работ;
* информация об источниках и способах финансирования;

|  |
| --- |
| * порядок выполнения и представления результатов создания информационной системы, частей системы или её отдельных частей. |
|  |
| * 1. **Описание объекта автоматизации** |

* общее описание объекта автоматизации;
* определение миссии и области деятельности организации;
* разработка контекстной диаграммы (в нотации IDEF0);
* ***Разработка структурной диаграммы*** *организации* (в среде MS Visio);
* определение целей использования системы (ограничения системы);
* информация об условиях эксплуатирования и характеристиках системы.

|  |
| --- |
| * 1. **Назначение и цели создания (модернизации) системы** |
| * виды деятельностей автоматизации; * список целей использования системы; * название и требуемые значения технических, технологических, экономических и других индикаторов, которые будут достигнуты с реализацией системы. |

* 1. **Требования к системе**
* *Общие требования к системе:*
  + требования к структуре и функционированию системы (список подсистем, иерархические уровни, степень централизации, способ обмена информации, режимы функционирования, взаимодействие с другими системами, перспективы развития);
  + требования к персоналу (роли, квалификация, режим работы, организация обучения, пользователи);
  + индикаторы связанные с назначением системы (адаптивность к изменениям системы управления и значений параметров, масштабируемость);
  + требования к надёжности, безопасности, эргономике, транспортабельности, эксплуатированию, техническому обслуживанию и рнмонтопригодности, защите от внешнего воздействия, авторскому праву, стандартизации и унификации.
* *Требования к функциональности (по подсистемам):*
  + перечень автоматизированных видов деятельностей;
  + временные рамки ревлизации каждой функциональности;
  + требования к качеству реализации каждой функциональности, форме представления выходов, точности и достоверности данных;
  + список и критерии отказов системы.
* *Требования к ресурсам:*
  + математическим – состав и сфера применения моделей и математических методов, новых и существующих алгоритмов;
  + информационным - состав, структура и организация данных, внутренний обмен данными, информационная совместимость с другими системами, используемые классификаторы, СУБД, контроль данных и использование массивов данных, процедуры придания юридической силы документов на выходе;
  + лингвистическим – языки программрования, языки взаимодействия пользователей с системой, системе кодирования, языки для входов и выходов;
  + программным – зависимость от платформы, качество и методы контроля, использование наборов алгоритмов и программ;
  + техническим;
  + метрологическим;
  + организационным – структура и функции отделов эксплуатации, защита от ошибочных действий;
  + методическим – номативно-техническая документация.
  1. **Структура и содержание работ по созданию системы**
* список стадий и этапов;
* сроки выполнения;
* список исполняющих организаций;
* способ и порядок экспертизы технической документации;
* программа по обеспечению надёжности;
* программа метрологического обеспечения.
  1. **Способ тестирования, верификации и приёма системы**
* типы, состав, объём и методы тестирования;
* требования по поэтапному приёму работ;
* положения приёмной комиссии.
  1. **Требования к структуре и содержанию работ по подготовке объекта автоматизации для вывода в эксплуатацию информационной системы**
* преобразование входной информации в машинную форму;
* изменения, внесённые в объект автоматизации;
* условия и порядок выбора и обучения персонала.
  1. **Требования к документации**
* представление разработанных документов;
* представление документации на носителях.

**Рекомендации к разработке курсовой работы**

**Курсовая работа должна содержать**:

1. Техническое задание;
2. Наработки из лабораторной работы: *№ 2 (Задание № 1, Задание № 2 и Задание № 3)*
3. Рисунок 2.2. Результативная контекстная диаграмма организации.
4. Рисунок 2.6. Функциональные блоки. Диаграмма декомпозиции первого уровня.
5. Рисунок 2.9. Диаграмма декомпозиции второго уровня “сборка и тестирование”.
6. Наработки из лабораторной работы: *№3*
7. Рисунок 3.5. Диаграмма декомпозиции первого уровня IDEF3.
8. Наработки из лабораторной работы: *№4*
9. Рисунок 4.7. Дочерняя диаграмма в нотации DFD
10. Наработки из лабораторной работы: *№6*
11. Рисунок 6.1. Логическая модель данных организации по сборке компьютеров и ноутбуков.

В рамках **курсовой работы** должны быть разработаны:

**“*Функциональная диаграмма”*** организации (в среде MS Visio).

**“*Диаграмма классов”***: описывает структуру системы путём выделения классов системы, их атрибутов и связей между классами (в языке UML).

**“*Диаграмма компонентов”***: описывает модуль, в котором система разделена на части (компоненты) и представляет зависимости между ними (в языке UML).

**“*Диаграмма деятельности”***: описывает последовательность операционных деятельностей компонентов системы (в языке UML);

**“*Диаграмма состояний”***: описывает состояния и переходные состояния системы (в языке UML);

**“*Диаграмма вариантов использования”***: описывает функциональности системы с точки зрения актёров, их цели и варианты использования (в языке UML).

**Элементы управления проектом**

Ряд взаимосвязанных мероприятий, которые производятся согласно плану достижения определённой цели / определённых результатов в заданный период времени; мероприятия завершаются в рамках проекта в тот момент, когда соответствующая цель была достигнута*.*

**Управление проектом** – деятельность, которая включает четыре функции управления проектом (1).

**Планирование**

- установка того, что должно быть выполнено;

- оценка необходимого количества веремени;

- оценка затрат.

**Организация**

- определение команды, которая будет работать над проектом;

- объединение руководителей, специалистов и пользователей.

**Контроль**

- мониторинг прогрессов и отчётов;

- сравнение планов с реальной ситуацией.

**Консультация**

- адаптирование проекта динамике организации;

- координирование людей для максимального их вовлечения в решение проекта.

При начале проекта известны лишь цели, которые должны быть выполнены и какие основные конечные результаты необходимо достичь, не перчисляя в деталях необходимые деятельности, их продолжительность и стоимость. Другими словами, на этом этапе проект ещё остаётся нераскрытым. Для создания более точного представления о проекте будет применён принцип “Разделяй и властвуй” для того, чтобы контролировать ещё неизвестные детали.

Применение этого принципа в управлении проектом находит место в процессе разложения работ, результатом которого является иерархическая структура работ (WBS - Work Breakdown Structure), которая является древовидной и отражает все работы, которые должны быть выполнены в рамках проекта.

Располагая всеми деталями мероприятий, можно произвести оценивание трудовых и материальных затрат для каждого мероприятия, используя все необходимые ресурсы: оборудование, аппаратуру, сырьё и материалы, людей и другие.

**Управление жизненным циклом проекта (PCM – Project Cycle Management) 6 этапов жизненного цикла проекта**

Проекты планируются и реализуются в последовательности этапов (5), начиная со стратегического документа согласованного развития, который определяет идею проекта в определённой области знаний, которая в дальнейшем формулируется, реализуется и в итоге проверена с целью улучшения стратегии и мероприятий для будущего развития.



Методы исследования

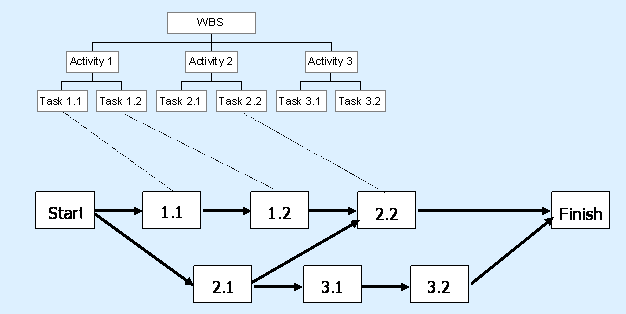
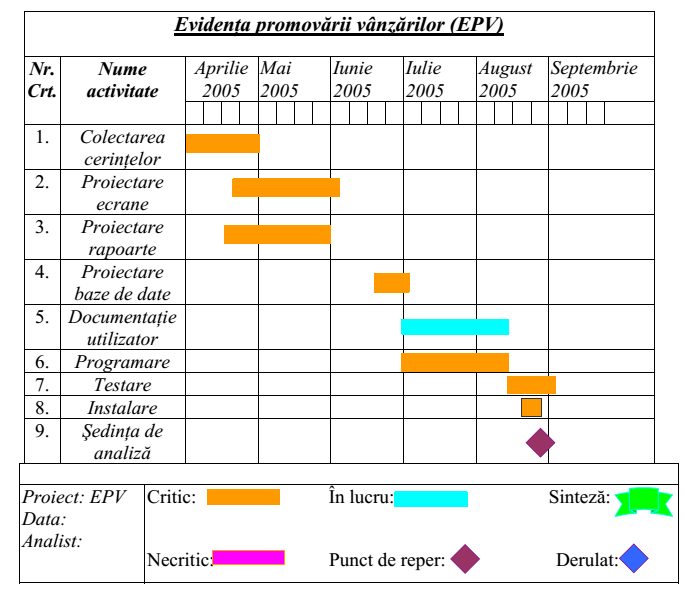


Диаграмма Ганта



**Примечание**

*Ответы на вопросы*

*Можно ли связать объект модели с внешними документами?*

Вы можете связать с объектом модели документацию, хранящуюся в формате приложения Windows, например, Word, Excel и т.д. Для этого необходимо использовать UDP свойство типа "Command". При задании значения этого свойства справа от имени свойства появляется кнопка ">". При щелчке по этой кнопке выполняется командная строка. Так если бы вы хотели связать объект с текстовым файлом, созданным в Word, то командная строка должно содержать следующее: [полный путь к файлу, начиная с имени диска] \ [имя файла].doc. Если вы нажмете кнопку ">", то откроется данный документ.

*Какие методологии используются при создании моделей в ERwin Process Modeler?*

В IDEF0 (первоначально SADT) система представляется как совокупность взаимодействующих работ или функций. Такая чисто функциональная ориентация является принципиальной - функции системы анализируются независимо от объектов, которыми они оперируют. Это позволяет более четко смоделировать логику и взаимодействие процессов организации.

Диаграммы потоков данных (Data flow diagramming, DFD) используются для описания документооборота и обработки информации.

Подобно IDEF0, DFD представляет модельную систему как сеть связанных между собой работ. 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, объекты, сотрудников или отделы, которые участвуют в обработке информации; внешние ссылки, которые обеспечивают интерфейс с внешними объектами, находящимися за границами моделируемой системы; таблицы для хранения документов.

IDEF 3 (Workflow diagram) - методология моделирования, использующая графическое описание информационных потоков, взаимоотношений между процессами обработки информации и объектов, являющихся частью этих процессов. С помощью этой методологии можно описывать сценарии действий сотрудников организации, например, последовательность обработки заказа или события, которые необходимо обработать за конечное время.

*По каким характеристикам происходит анализ качества созданной модели?*

ERwin Process Modeler предоставляет аналитику два инструмента для оценки модели - стоимостный анализ, основанный на работах (Activity Based Costing, ABC), и свойства, определяемые пользователем (User Defined Properties, UDP). ABC является широко распространенной методикой, используемой международными корпорациями и государственными организациями для идентификации истинных движителей затрат в организации. При проведении функционально стоимостного анализа используются единицы единиц денег и времени, которые задаются разработчиком. Если стоимостных показателей недостаточно, имеется возможность внесения собственных метрик - свойств, определенных пользователем. Имеется возможность создания 18 различных типов UDP (таких как текст, командная строка, массив строк, дата и т.д.).

*Какие отчеты можно создавать в AllFusion Process Modeler?*

Документирование результатов моделирования является важнейшей задачей при проектировании ИС, причем актуальность этой задачи тем больше, чем масштабнее проект. Для создания документации на основе модели процессов или модели данных можно воспользоваться как встроенными средствами AllFusion PM и AllFusion Erwin DM, так и внешними генераторами отчетов. Каждый из этих способов имеет свои достоинства и недостатки. Здесь мы рассмотрим встроенные возможности AllFusion PM:

Model report - Этот отчет включает информацию о контексте модели - имя модели, точку зрения, область, цель, имя автора, дату создания и др.

Diagram report - Отчет по конкретной диаграмме. Включает список объектов (работ, стрелок, хранилищ данных, внешних ссылок).

Diagram object report - Наиболее полный отчет по модели. Может включать полный список объектов модели (работ, стрелок с указанием их типа и др.) и свойства, определяемые пользователем.

Activity cost report - Отчет о результатах стоимостного анализ.

Arrow report - Отчет по стрелкам. Может содержать информацию из словаря стрелок, информацию о работе-источнике, работе-назначении стрелки и информацию о разветвлении и слиянии стрелок.

DataUsage report - Отчет о результатах связывания модели процессов и модели данных.

Model Consistency report - Отчет, содержащий список синтаксических ошибок модели.

Where Used report - Показывает, на каких диаграммах используются те или иные объекты.

Report Template Builder является простым генератором отчетов, который входит в состав AllFusion PM.

*Что такое модель AS-IS и модель ТО-ВЕ?*

Построение модели AS-IS (как есть) производится на основе существующей документации, анкетирования, опроса служащих предприятия и др. источников. Полученная модель служит для выявления неуправляемых работ, работ, не обеспеченных ресурсами и других недостатков в организации деятельности предприятия. Исправление недостатков, перенаправление информационных и материальных потоков приводит к созданию модели TO-BE (как будет). Как правило, строится несколько моделей ТО-ВЕ, среди которых определяют наилучший вариант. Распространенная ошибка при создании модели AS-IS - создание идеализированной модели. Пример - модель созданная на основе знаний руководителя, а не конкретного исполнителя работ. В результате получается искаженная модель, такая модель несет название SHOULD BE (как должно бы быть). Технология проектирования ИС подразумевает сначала создание модели AS-IS, ее анализ и улучшение бизнес-процессов, т.е. создание модели ТО-ВЕ, и только на основе модели ТО-ВЕ строится модель данных, прототип и затем окончательный вариант ИС. Построение системы на основе модели AS-IS приводит к автоматизации редприятия по принципу "все оставить как есть, только чтобы компьютеры стояли", т.е. ИС автоматизирует несовершенные бизнес-процессы и дублирует, а не заменяет существующий документооборот. В результате внедрение и эксплуатация такой системы приводит лишь к дополнительным издержкам на закупку оборудования, создание программного обеспечения и сопровождения того и другого. Иногда текущая модель AS-IS и будущая ТО-ВЕ различаются очень сильно, так что переход от начального состояния к конечному становится неочевидным. В этом случае необходима третья модель, описывающая процесс перехода от начального к конечному состоянию системы, поскольку такой переход - это тоже бизнес - процесс.

*Как добавить Referent Tool, External Reference Tool и Data Store Tool на диаграмму?*

Для того, чтобы вставить эти объекты необходимо:

1. Открыть диаграмму на которую необходимо произвести вставку.

2. Выбрать нужный объект в палитре инструментов.

3. Кликнуть левой клавишей мышки на свободной области диаграммы.

4. Ввести имя объекта.

5. Перейти на закладку "Objects" навигатора "Model Explorer".

6. Найти вставленный элемент.

7. Перетаскиванием добавить объект на диаграмму.

*Как русифицировать словари и модели?*

Для того, чтобы русифицировать словари и тексты, вводимые пользователем в AllFusion Process Modeler 7.2 (BPWin), необходимо:

1. Запустить файл импорта информации в реестр: Font\_RUS.reg .

2. На вопрос о добавлении информации в реестр ответить положительно.

3. Перезагрузить систему.

*Как изобразить стрелку "выход обратная связь на вход" к тому же функциональному блоку?*

BPWin не позволяет создать циклическую обратную связь. Однако если такая необходимость возникает, то необходимо: Создать обычную связь по входу. Затем разветвлить стрелку, направить новую ветвь обратно ко входу работы-источника. Удалить старую ветвь строки выхода.

*Как совместить на одной картинке две модели?*

Чтобы сохранить графический файл с совмещенными моделями, выбираем Visual Diagram Difference из меню Tools. В открывшемся диалоговом окне необходимо выбрать модели для совмещения и нажать Compare. Так же с помощью этого инструмента можно получить графический файл с одной моделью, для этого в окне Visual Diagram Difference отмечаем одну и ту же модель.    Название функции выходит за рамки функционального блока. Что делать?    Чтобы текст располагался внутри функциональных блоков с переносами по словам, а не в одну строчку, следует 1.  настроить свойства модели: меню Model/Model Properties, закладка Layout, в разделе "Fit Name in Box" выбрать "Wrap text to fit box"; 2.  изменить ширину функционального блока на диаграмме. Если вам не нравится автоматический вариант расстановки переносов слов в названиях функций, стрелок и т.п., вы можете расставить переносы принудительно в соответствующем словаре (меню Dictionary).

*Как активизировать пункт меню File\Export\ERwin для экспорта данных из Process Modeler в ERwin?*

В Process Modeler в словаре сущностей и в словаре атрибутов установите флажок "Exchange with ERwin". После этого пункт меню File\Export\ERwin станет активным.

При переходе с IDEF0 на DFD граничные стрелки мигрируют на DFD. Нотация DFD не разрешает граничные стрелки. Что с ними делать?

Действительно, нотация DFD не разрешает граничные стрелки - все стрелки должны начинаться и заканчиваться на работах, хранилищах данных или внешних сущностях. Чтобы строго следовать правилам нотации DFD, следует: удалить все 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 затонеллировать Строго придерживаться правил нотации DFD при создании смешанных моделей не всегда удобно, поэтому Process Modeler позволяет создавать граничные стрелки на диаграммах DFD и не идентифицирует такие стрелки как синтаксическую ошибку.

*Почему после экспорта данных из ERwin в модели Process Modeler ничего нового не появляется?*

Экспортированные из ERwin данные следует искать не на диаграммах, а в словарях Process Modeler (см. меню Dictionary\Entity и Dictionary\Attribute).

*Может ли BPWin автоматически указывать номер дочерней диаграммы на родительском функциональном блоке?*

В методологии IDEF0 указано, что в правом нижнем угле функционального блока при наличии декомпозиции должен указываться номер дочерней диаграммы. Эта возможность поддерживается Process Modeler`ом автоматически после включения соответствующих настроек отображения модели: Включить отображение номера функционального блока: Model\Model Properties\Display\Activity Numbers. Выбрать подходящий способ отображения номера блока: Model\Model Properties\Numbering. После такой настройки вы сможете увидеть, что каждый функциональный блок имеет номер в правом нижнем углу, и каждая дочерняя диаграмма имеет такой же номер в подвале рамки в поле NODE (Узел - для русифицированной рамки). Чтобы визуально определять, какие из функциональных блоков диаграммы имеют дочерние диаграммы, а какие - нет, требуется включить еще одну опцию: Model\Model Properties\Display\Leaf Corners. После этого вы увидите разницу в отображении блоков, имеющих декомпозицию и блоков, не имеющих декомпозиции: в левом верхнем углу блока без декомпозиции нарисована диагональная черта. Таким образом, в Process Modeler вы можете визуально определить, есть ли расшифровка у блока и если есть, то номер детализирующей диаграммы.

*Есть ли отличие в русификации рамки диаграммы в Process Modeler 7?*

Небольшое отличие есть. Требуется скопировать русифицированный файл BPwinLoc.INI в каталог \Windows (или \winnt) до начала инсталляции продукта. Если вы уже установили продукт без русификации рамки или хотите изменить параметры настройки рамки, то придется править реестры Windows. Для этого зайдите в редактор реестров Windows (Start\Run\ regedit) и исправьте значения полей в HKEY\_CURRENT\_USER\Software\ComputerAssociates\AFPMr7\AllFusion Process Modeler\BPwinLoc по своему усмотрению (или в соответствии с содержимым модифицированного файла BPwinLoc.INI), далее перезагрузите Windows или закройте сеанс работы с Windows.

*Открыл ранее созданную в BPwin модель, не могу сохранить в ней новые изменения, в чем причина?*

Вы запускаете модель двойным нажатием кнопки мышки. По умолчанию для расширения \*.bp1 у вас запускается не Process Modeler, а другой продукт пакета - AllFusion Model Navigator. Navigator, действительно, не позволяет сохранять изменения модели - только просматривать и создавать отчеты. Решение 1: запустите сначала Process Modeler, а в нем откройте модель, которую хотите изменять. Решение 2: смените программу, запускаемую по умолчанию для расширения \*.bp1, на AllFusion Process Modeler. Как назначить автора конкретной диаграмме?   Чтобы назначить автора конкретной диаграмме, нужно назначить автора для родительской функции на родительской диаграмме: правая кнопка мышки на родительскую функцию, закладка Name, пункт Author.

*В чем принципиальное отличие Process Modeler от "рисовалок" типа MS Visio?*

Основные отличия: Process Modeler не позволяет отойти от выбранного стандарта (IDEF0, IDEF3, DFD): выбрав стандарт (IDEF0, IDEF3 или DFD) вы работаете только с набором графических элементов данного стандарта; можно автоматически проверить синтаксис диаграммы. Информация вводится/изменяется один раз в одном месте. Process Modeler интегрируется с ERwin Data Modeler, поддерживая синхронизацию моделей процессов и моделей данных. Process Modeler предоставляет возможность автоматической генерации SwimLane-диаграмм (графический прообраз должностных инструкций) из IDEF3-диаграмм. При декомпозиции диаграмм Process Modeler сохраняет !!! контекст: стрелки, касающиеся декомпозируемого блока, мигрируют на диаграмму декомпозиции. Поэтому декомпозиция функции начинается не с нуля, а с наличия контекста в виде граничных стрелок, мигрировавших из родительской диаграммы. Кроме этого инструмент Туннелей позволяет автоматически отслеживает дисбаланс стрелок у родительской функции и в дочерней диаграмме. Process Modeler позволяет генерировать произвольные отчеты по модели.

*Как получить отчет "роль (должность)-группа ролей (отделы)"?*

Используйте Report Template Builder. Создайте новый шаблон отчета, из раздела Available Sections перетащите в создаваемый шаблон (Report Layout) раздел Diagram Role. В свойствах Diagram Role нужно выбрать "Role", "Group", далее выбрать удобный формат отчета (HTML, RTF, TXT или PDF) и запустить генерацию отчета. Как русифицировать рамку (форму) диаграммы в Process Modeler? В ERwin Process Modeler 7.2.8 появилась возможность настроить рамку диаграммы, включая возможность заменить английские названия полей рамки на собственные названия (например, на русские названия полей согласно рекомендациям Р 50.1.028-2001), а также изменить положение вертикальных линий рамки. Для настройки рамки диаграммы надо модифицировать файл BPwinLoc.INI, расположенный в каталоге \Windows (или \winnt). Вы можете просто скачать уже модифицированный файл BPwinLoc.INI. Скачать ERwin Process Modeler 7.2.8 можно здесь.

Когда запускаю Process Modeler в Win XP SP2 под обычным юзером, выдается ошибка "Unable to locate configuration file for ModelMart interface…". Что делать?

AllFusion Process Modeler (ранее BPwin) корректно работает под Windows XP sp2. Следует помнить два момента: После установки продукта под администратором продукт надо запустить и закрыть; При работе с продуктом под обычным пользователем нужно помнить, что сохранять модели можно только в разрешенные папки.

*После работы вируса в Process Modeler исчезли некоторые панели инструментов. Что делать?*

Первое, что надо попробовать - зайти в меню View и проставить чеки для требуемых панелей. Если не помогает, то удалить BPwin41.ini.

*Как подключить к Activity документ, который лежит в интернете по определенному адресу?*

Присоединение документов к работам и стрелкам в Process Modeler настраивается с помощью UPD - свойств, определенных пользователем. Для этого нужно создать UDP типа Command или Command List и подключить UDP к требуемой работе или стрелке. В качестве значения Value для UPD указать адрес Интернет-страницы, включая название протокола, например, http://inc.istu.ru. Если необходимо, в Value указать полную командную строку с явным указанием приложения, требуемого для открытия документа, например, explorer.exe http://inc.istu.ru. Как удалить роль из activity? По правой кнопке мышки щелкнуть по activity, выбрать Roles, в открывшемся диалоге встать на роль, которую нужно удалить и нажать клавишу "Del".

Источник: <http://pro-spo.ru/information-required-to-install/1043-erwin-process-modeler-bpwin>

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. Unelte şi tehnici de dezvoltare a sistemelor**,** Prof.dr.ing. Liviu Roşca
2. AllFusion Process Modeler. 2002 Computer Associates International, Inc. (CA)
3. CA ERwin Data Modeler.Implementation Guide. Release 9.64.00

## IDEF3 Representation of Process Workflows. http://www.atl.external.lmco.com/projects/rassp/RASSP\_legacy/appnotes/METHOD/APNOTE\_METHOD\_03.HTM

1. IDEF1X Introduction. Derek Asirvadem. 1993-2013 Software Gems Pty Ltd
2. Ovidiu S. Noran, 2000. Griffith University. School of Computing and Information Technology. Domain: Modelling Languages. Business Modelling: UML vs. IDEF, <http://www.cit.gu.edu.au/noran>
3. [www.mce.utcluj.ro/.../Managementul%20proiectelor\_curs2.ppt](http://www.mce.utcluj.ro/.../Managementul%20proiectelor_curs2.ppt)