

Лабораторная работа № 1. Понятие информационного процесса. Виды информационных процессов (4 часа)

VRwin позволяет аналитику создавать сложные модели бизнес-процессов при минимальных усилиях. VRwin поддерживает три методологии - IDEF0, IDEF3 и DFD. Каждая из них призвана решать свои специфические задачи. Также можно строить смешанные модели. Модель в VRwin рассматривается как совокупность работ, каждая из которых оперирует с некоторым набором данных. Работы изображаются в виде прямоугольников (блоков), данные - в виде стрелок (дуг).

Основу методологии IDEF0 составляет графический язык описания бизнес-процессов. Модель в IDEF0 представлена совокупностью иерархически упорядоченных и логически связанных диаграмм. Каждая диаграмма располагается на отдельном листе. Можно выделить четыре типа диаграмм:

- контекстную диаграмму A-0 (в каждой модели может быть только одна контекстная диаграмма);
- диаграммы декомпозиции (в том числе диаграмма первого уровня декомпозиции A0, раскрывающая контекстную);
- диаграммы дерева узлов;
- диаграммы только для экспозиции (FEO).

Контекстная диаграмма является вершиной древовидной структуры диаграмм и представляет собой самое общее описание системы и ее взаимодействия с внешней средой (как правило, здесь описывается основное назначение моделируемого объекта). После описания системы в целом проводится разбиение ее на крупные фрагменты. Этот процесс называется функциональной декомпозицией, а диаграммы, которые описывают каждый фрагмент и взаимодействие фрагментов, называются диаграммами декомпозиции. После декомпозиции контекстной диаграммы (т.е., получения диаграммы A0) проводится декомпозиция каждого блока диаграммы A0 на более мелкие фрагменты и так далее, до достижения нужного уровня подробности описания. После каждого сеанса декомпозиции проводятся сеансы экспертизы - эксперты предметной области (обычно это интервьюируемые аналитиками сотрудники предприятий) указывают на соответствие реальных бизнес-процессов созданным диаграммам. Найденные несоответствия исправляются, и только после прохождения экспертизы без замечаний можно приступать к следующему сеансу декомпозиции. Так достигается соответствие модели реальным бизнес-процессам на любом и каждом уровне модели. Синтаксис описания системы в целом и каждого ее фрагмента одинаков во всей модели. Диаграмма дерева узлов показывает иерархическую зависимость работ, но не взаимосвязи между работами. Диаграмм деревьев узлов может быть в модели сколько угодно, поскольку дерево может быть построено на произвольную глубину и не обязательно с корня.

Диаграммы для экспозиции (FEO) строятся для иллюстрации отдельных фрагментов модели, для иллюстрации альтернативной точки зрения, либо для специальных целей.

Каркас диаграммы.

На рис.1 показан типичный пример контекстной диаграммы с граничными рамками, которые называются каркасом диаграммы. Каркас содержит заголовок (верхняя часть рамки, табл.3) и подвал (нижняя часть, табл.4). Заголовок каркаса используется для отслеживания диаграммы в процессе моделирования. Нижняя часть используется для идентификации и позиционирования в иерархии диаграмм. Значения полей каркаса задаются в диалоге Diagram Properties (в меню Edit/Diagram Properties).

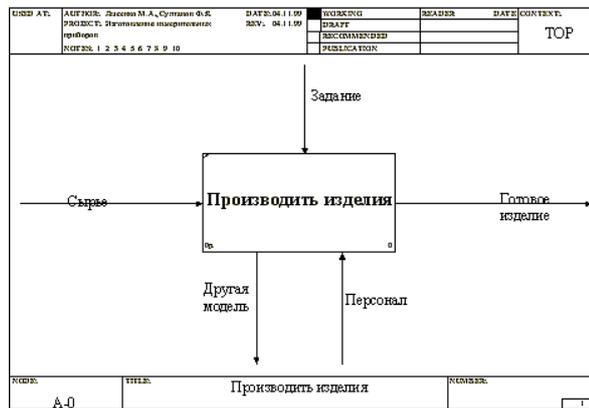


Рис.1.Контекстная диаграмма

Поля заголовка каркаса (слева направо)

Табл. 3

| Поле | Смысл |
|----------------------------|--|
| Used At | Используется для указания на родительскую работу в случае, если на текущую диаграмму ссылались посредством стрелки вызова. |
| Author, Date, Rev, Project | Имя создателя диаграммы, дата создания и имя проекта, в рамках которого была создана диаграмма. REV - дата последнего редактирования диаграммы. |
| Notes 1 2 3 4 5 6 7 8 9 10 | Используется при проведении сеанса экспертизы. Эксперт должен (на бумажной копии диаграммы) указать число замечаний, вычеркивая цифру из списка каждый раз при внесении нового замечания. |
| Status | Статус отображает стадию создания диаграммы, отображая все этапы публикации. |
| Working | Новая диаграмма, кардинально обновленная диаграмма или новый автор диаграммы. |
| Draft | Диаграмма прошла первичную экспертизу и готова к дальнейшему обсуждению. |
| Recommended | Диаграмма и все ее сопровождающие документы прошли экспертизу. Новых изменений не ожидается. |
| Publication | Диаграмма готова к окончательной печати и публикации. |
| Reader | Имя читателя (эксперта). |
| Date | Дата прочтения (экспертизы). |
| Context | Схема расположения работ в диаграмме верхнего уровня. Работа, являющаяся родительской, показана темным прямоугольником, остальные - светлым. На контекстной диаграмме (А-0) показывается надпись TOP. В левом нижнем углу показывается номер по узлу родительской диаграммы. |

Поля подвала каркаса (слева направо)

Табл. 4

| Поле | Смысл |
|--------|--|
| Node | Номер узла диаграммы (номер родительской работы) |
| Title | Имя диаграммы. По умолчанию - имя родительской работы |
| Number | C-Number, уникальный номер версии диаграммы |
| Page | Номер страницы, может использоваться как номер страницы при формировании |

| |
|-------|
| папки |
|-------|

Задание.

На основе резюме, описывающих функционирование конкретного отдела РГУ нефти и газа им.И.М.Губкина, создать контекстную диаграмму А-0. Выделить основные его функции и создать диаграмму А0. Разбить каждую функцию на подфункции и диаграммы третьего уровня. Предоставить иерархию диаграмм.

Вопросы.

1. Каковы стадии жизненного цикла информационных систем, их основное содержание?
2. Что такое реинжиниринг бизнес-процессов?
3. Какие виды работ рекомендуется выполнить при построении моделей деятельности, какие средства и методологии при этом используются?
4. Каковы основные функции CASE-средства VPwin?
5. Как представляется функциональная модель деятельности в методологии IDEF0?

Лабораторная работа № 2.

Стандарты разработки ИС. При-менение современных CASE-средств для разработки информа-ционных систем (4 часа)

Работы обозначают поименованные процессы, функции или задачи, которые происходят в течение определенного времени и имеют распознаваемые результаты. Работы изображаются в виде прямоугольников (блоков). Все работы должны быть названы и определены. Имя работы должно быть глаголом (например, "Изготовить деталь", "Принять заказ" и т.д.). Работу можно добавить в диаграмму, щелкнув по кнопке  на палитре инструментов, а затем по свободному месту на диаграмме. Работы на диаграммах декомпозиции располагаются по диагонали от левого верхнего угла к правому нижнему (рис.2). Такой порядок называется порядком доминирования. Согласно этому принципу расположения в левом верхнем углу располагается самая важная работа или работа, выполняемая по времени первой. Далее вправо вниз располагаются менее важные или выполняемые позже работы.

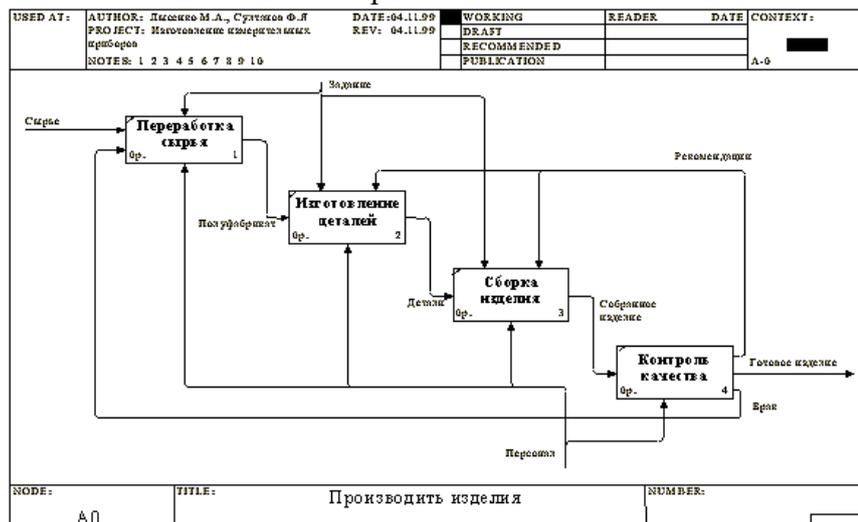


Рис.2. Диаграмма декомпозиции

Для внесения имени работы следует щелкнуть по работе правой кнопкой мыши, выбрать в меню пункт Name Editor и в появившемся диалоге внести имя работы (рис.3). Диаграммы декомпозиции содержат родственные работы, т.е. дочерние работы, имеющие общую родительскую работу.

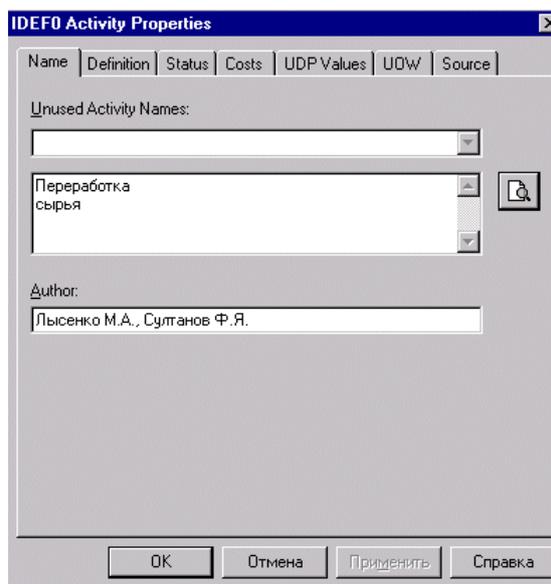


Рис.3.Внесение имени работы

Для создания диаграммы декомпозиции следует щелкнуть по кнопке  и выбрать на диаграмме работу, которую необходимо декомпозировать. Возникает диалог Activity Box Count (рис.4), в котором следует указать нотацию новой диаграммы. Надо выбрать IDEF0 и надавить ОК.



Рис.4.Выбор нотации диаграммы

На диаграмме декомпозиции работы нумеруются автоматически слева направо. Номер работы показывается в правом нижнем углу. В левом верхнем углу изображается небольшая диагональная черта, которая показывает, что данная работа не была декомпозирована.

Стрелки (Arrows).

Взаимодействие работ с внешним миром описывается в виде стрелок. Стрелки представляют собой некую информацию и именуются существительными (например, "Заготовка", "Изделие", "Заказ").

В IDEF0 различают пять типов стрелок.

- Вход (Input) - материал или информация, которая используется или преобразуется работой для получения результата (выхода). Допускается, что работа может не иметь ни одной стрелки входа. Каждый тип стрелок подходит к определенной стороне блока, или выходит из нее. Очень часто сложно определить, 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.

- Управление (Control) - правила, стратегии, процедуры или стандарты, которыми руководствуется работа. Каждая работа должна иметь хотя бы одну стрелку управления. Управление влияет на работу, но не преобразуется ей. Если цель работы - изменить процедуру или стратегию, то такая процедура или стратегия будет для работы входом.

- · Выход (Output) - материал или информация, которые производятся работой. Каждая работа должна иметь хотя бы одну стрелку выхода. Работа без результата не имеет смысла.

- · Механизм (Mechanism) - ресурсы, которые выполняют работу, например персонал предприятия, станки, устройства и т.д.

- · Вызов (Call) - специальная стрелка, указывающая на другую модель работы. Рисуеться как исходящая из нижней грани работы. Стрелка вызова используется для указания того, что некоторая работа выполняется за пределами моделируемой системы. Используются в механизме слияния и разделения моделей.

Каждый тип стрелок подходит к определенной стороне блока, или выходит из нее.

Стрелка входа рисуется как входящая в левую грань работы. Стрелка управления рисуется как входящая в верхнюю грань. Выход рисуется как исходящая стрелка из правой грани. Механизм - входит в нижнюю.

Граничные стрелки.

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 надо:

- · щелкнуть по кнопке с символом стрелки  в палитре инструментов. Дальше перенести курсор к левой стороне экрана, пока не появится начальная штриховая полоска;

- · щелкнуть один раз по полоске (откуда выходит стрелка) и еще раз в левой части работы со стороны входа (где заканчивается стрелка);

- · вернуться в палитру инструментов и выбрать опцию редактирования стрелки 

- · щелкнуть правой кнопкой мыши на линии стрелки, во всплывающем меню выбрать пункт Name Editor и добавить имя стрелки в закладке Name диалога IDEF0 Arrow Properties.

Стрелки управления, входа, 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 штриховая полоска, и щелкнуть один раз по ней. Имена вновь внесенных стрелок автоматически заносятся в словарь (Arrow Dictionary).

Словарь стрелок (Arrow Dictionary) редактируется при помощи специального редактора Arrow Dictionary Editor (рис.5), в котором определяется стрелка и вносятся 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.

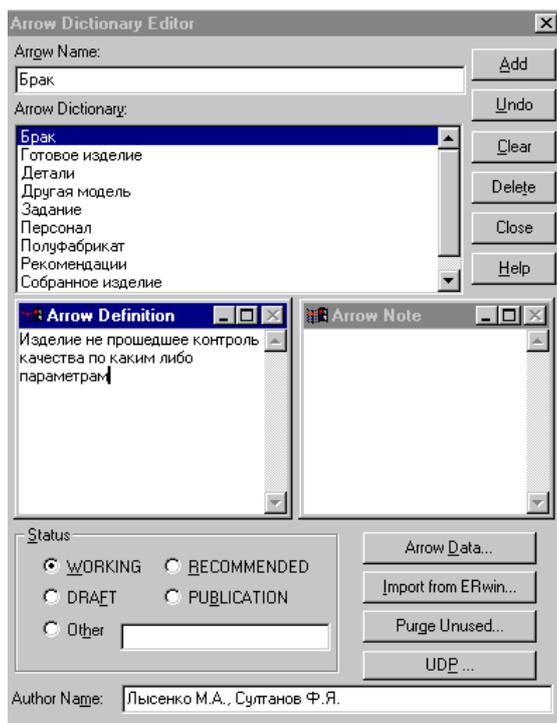


Рис.5.Редактор словаря стрелок

Поскольку формальные определения часто сложны для восприятия, аналитик вынужден употреблять профессиональный жаргон, а чтобы не возникало неоднозначных трактовок, в словаре стрелок каждому понятию можно дать расширенное и, если это необходимо, формальное определение.

Внутренние стрелки.

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;
- · 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 вышестоящей;
- · обратная связь по входу (output-input feedback), когда выход нижестоящей работы направляется на вход вышестоящей. Такая связь, как правило, используется для описания циклов;
- · обратная связь по управлению (output-control feedback), когда выход нижестоящей работы направляется на управление вышестоящей. Обратная связь по управлению часто свидетельствует об эффективности бизнес-процесса;
- · 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.

Явные стрелки.

Явная стрелка имеет источником одну-единственную работу и назначением тоже одну-единственную работу.

Разветвляющиеся и сливающиеся стрелки.

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 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.

Тоннелирование стрелок.

Вновь внесенные граничные стрелки на диаграмме декомпозиции нижнего уровня изображаются в квадратных скобках и автоматически не появляются на диаграмме верхнего уровня. Для их "перетаскивания" вверх нужно сначала выбрать кнопку  на палитре инструментов и щелкнуть по квадратным скобкам граничной стрелки. Появится диалог Border Arrow Editor (рис.6).



Рис.6.Диалог для тоннелирования стрелок

Если щелкнуть по кнопке Resolve Border Arrow, стрелка мигрирует на диаграмму верхнего уровня, если по кнопке Change To Tunnel - стрелка будет затоннелирована и не попадет на другую диаграмму. Тоннельная стрелка изображается с круглыми скобками на конце.

Тоннелирование 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. 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 тоннелирование стрелки на самом нижнем уровне. Такое тоннелирование называется "Не-в-родительской-диаграмме". Другим примером тоннелирования 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. В этом случае стрелка механизма на нижнем уровне может быть удалена, после чего на родительской диаграмме она может быть затоннелирована ("Не-в-дочерней-работе").

Задание.

Исходя из результатов предыдущей лабораторной работы, создать все диаграммы в программе, расположить на них все блоки и дуги, описывающие заданный отдел. Получить законченную модель функционирования отдела.

Вопросы.

1. Что такое CASE-технологии, их достоинства и преимущества?
2. Проведите сравнительный анализ традиционной технологии разработки и разработки с помощью CASE-технологии.
3. Каковы основные объекты диаграмм функциональной модели по методологии IDEF0?

4. Что обозначают работы в диаграммах функциональной модели, как они отображаются по методологии IDEF0?
5. Для чего предназначены стрелки в диаграммах функциональной модели, каковы их типы и виды?
6. Для чего предназначен словарь стрелок?
7. Каковы типы связей работ по методологии IDEF0?
8. Что такое тоннелирование стрелок, для чего оно нужно, каковы виды тоннелирования?

Лабораторная работа № 3. Логическая, физическая и функциональная модели данных ИС (4 часа)

ERwin - средство концептуального моделирования БД, использующее методологию IDEF1X. ERwin реализует проектирование схемы БД, генерацию ее описания на языке целевой СУБД (ORACLE, Informix, Ingres, Sybase, DB/2, Microsoft SQL Server, Progress и др.) и реинжиниринг существующей БД. ERwin выпускается в нескольких различных конфигурациях, ориентированных на наиболее распространенные средства разработки приложений 4GL. Версия ERwin/OPEN полностью совместима со средствами разработки приложений PowerBuilder и SQLWindows и позволяет экспортировать описание спроектированной БД непосредственно в репозитории данных средств. Для ряда средств разработки приложений (PowerBuilder, SQLWindows, Delphi, Visual Basic) выполняется генерация форм и прототипов приложений. Сетевая версия ERwin ModelMart обеспечивает согласованное проектирование БД и приложений в рабочей группе.

Основные получаемые преимущества:

- существенное повышение скорости разработки за счет мощного редактора диаграмм, автоматической генерации базы данных, автоматической подготовки документации;
- нет необходимости ручной подготовки SQL-предложений для создания базы данных;
- возможность легко вносить изменения в модель при разработке и расширении системы;
- возможность автоматической подготовки отчетов по базе данных; важно, что эти отчеты всегда в точности соответствуют реальной структуре БД;
- разработчики прикладного программного обеспечения снабжены удобными в работе диаграммами;
- тесная интеграция со средствами 4GL позволяет уже на стадии информационного моделирования задавать отображение данных в приложениях;
- обратное проектирование позволяет документировать и вносить изменения в существующие информационные системы;
- поддержка однопользовательских СУБД позволяет использовать для персональных систем современные технологии, что значительно упрощает переход от настольных систем к системам в технологии клиент-сервер (upsizing).

Построение моделей в ERwin

Возможны две точки зрения на информационную модель и, соответственно, два уровня модели. Первый - логический уровень (точка зрения пользователя) означает прямое отображение фактов из реальной жизни. Например, люди, столы, отделы, собаки и компьютеры являются реальными объектами. Они именуется на естественном языке, с любыми разделителями слов (пробелы, запятые и т.д.). На физическом уровне модели рассматривается использование конкретной СУБД, определяются типы данных (например, целое или вещественное число), индексы для таблиц. ERwin предоставляет возможности создавать и управлять этими двумя различными уровнями представления одной диаграммы (модели), равно как и иметь много вариантов

отображения на каждом уровне. Термин "логический уровень" в ERwin соответствует концептуальной модели.

Этапы построения информационной модели:

- определение сущностей;
- определение зависимостей между сущностями;
- задание первичных и альтернативных ключей;
- определение атрибутов сущностей;
- приведение модели к требуемому уровню нормальной формы;
- переход к физическому описанию модели: назначение соответствий имя сущности - имя таблицы, атрибут сущности - атрибут таблицы;
- задание триггеров, процедур и ограничений;
- генерация базы данных.

Erwin создает визуальное представление (модель данных) для решаемой задачи. Это представление может использоваться для детального анализа, уточнения и распространения документации, необходимой в цикле разработки. Однако ERwin далеко не только инструмент для рисования. ERwin автоматически создает базу данных (таблицы, индексы, хранимые процедуры, триггеры для обеспечения ссылочной целостности и другие объекты, необходимые для управления данными).

Создание сущности.

Для внесения сущности в модель необходимо щелкнуть по кнопке сущности на панели инструментов (Erwin Toolbox) , затем - по тому месту на диаграмме, где необходимо расположить новую сущность. Щелкнув правой кнопкой мыши по сущности и выбрав из всплывающего меню пункт Entity Editor, можно вызвать диалог Entity Editor, в котором определяются имя, описание и комментарии сущности. Каждая сущность должна быть полностью определена с помощью текстового описания в закладке Definition. Эти определения полезны как на логическом уровне, поскольку позволяют понять, что это за объект, так и на физическом уровне, поскольку их можно экспортировать как часть схемы и использовать в реальной БД (CREATE COMMENT on entity_name). Закладки Note, Note2, Note3, UDP (User Defined Properties - Свойства, определенные пользователем) служат для внесения дополнительных комментариев и определений к сущности.

В закладке Icon каждой сущности можно поставить в соответствие изображение, которое будет отображаться в режиме просмотра модели на уровне иконок и изображение, которое будет отображаться на всех других уровнях.

Закладка UDP диалога Entity Editor служит для определения свойств, определяемых пользователем (User - Defined Properties). При нажатии на кнопку  этой закладки вызывается диалог User - Defined Property Editor (также вызывается из меню Edit/UDPs). В нем необходимо указать вид объекта, для которого заводится UDP (диаграмма в целом, сущность, атрибут и т.д.) и тип данных. Для внесения нового свойства следует щелкнуть в таблице по кнопке  и внести имя, тип данных, значение по умолчанию и определение.

Создание атрибутов.

Для описания атрибутов следует, щелкнув правой кнопкой по сущности, выбрать в появившемся меню пункт Attribute Editor. Появится диалог Attribute Editor. Если щелкнуть по кнопке New, то в появившемся диалоге New Attribute можно указать имя атрибута, имя соответствующей ему в физической модели колонки и домен. Домен атрибута будет использоваться при определении типа колонки на уровне физической модели.

Для атрибутов первичного ключа в закладке General диалога Attribute Editor необходимо сделать пометку в окне выбора Primary Key. Закладки Definition, Note и UDP несут те же функции, что и при определении сущности, но на уровне атрибутов.

Для большей наглядности диаграммы каждый атрибут можно связать с иконкой. Это

можно сделать при помощи списка выбора Icon в закладке General. Очень важно дать атрибуту правильное имя. Атрибуты должны именоваться в единственном числе и иметь четкое смысловое значение. Согласно синтаксису IDEF1X, имя атрибута должно быть уникальным в рамках модели (а не только в рамках сущности!). По умолчанию при попытке внесения уже существующего имени атрибута ERwin переименовывает его. Например, если атрибут Комментарий уже существует в модели, другой атрибут (в другой сущности) будет назван Комментарий/2, затем Комментарий/3 и т.д. При переносе атрибутов внутри и между сущностями можно воспользоваться техникой drag&drop, выбрав кнопку  в палитре инструментов.

Создание связи.

Для создания новой связи следует выбрать идентифицирующую или неидентифицирующую связь в палитре инструментов (ERwin Toolbox), щелкнуть сначала по родительской, а затем по дочерней сущности.

В палитре инструментов кнопка  соответствует идентифицирующей связи, кнопка 

связи многие-ко-многим и кнопка  соответствует неидентифицирующей связи. Для редактирования свойств связи следует щелкнуть правой кнопкой мыши по связи и выбрать на контекстном меню пункт Relationship Editor.

В закладке General появившегося диалога можно задать мощность, имя и тип связи.

Мощность связи (Cardinality) - служит для обозначения отношения числа экземпляров родительской сущности к числу экземпляров дочерней.

Различают четыре типа мощности: общий случай, когда одному экземпляру родительской сущности соответствуют 0, 1 или много экземпляров дочерней сущности, не помечается каким-либо символом; символом P помечается случай, когда одному экземпляру родительской сущности соответствуют 1 или много экземпляров дочерней сущности (исключено нулевое значение);

символом Z помечается случай, когда одному экземпляру родительской сущности соответствуют 0 или 1 экземпляр дочерней сущности (исключены множественные значения);

цифрой помечается случай, когда одному экземпляру родительской сущности соответствует заранее заданное число экземпляров дочерней сущности.

По умолчанию символ, обозначающий мощность связи, не показывается на диаграмме. Для отображения имени следует в контекстном меню, которое появляется, если щелкнуть правой кнопкой мыши по любому месту диаграммы, не занятому объектами модели, выбрать пункт Display Options/Relationship и затем включить опцию Cardinality.

Тип связи (идентифицирующая/неидентифицирующая).

В IDEF1X различают зависимые и независимые сущности. Тип сущности определяется ее связью с другими сущностями. Идентифицирующая связь устанавливается между независимой (родительский конец связи) и зависимой (дочерний конец связи) сущностями. Когда рисуется идентифицирующая связь, ERwin автоматически преобразует дочернюю связь в зависимую. Зависимая сущность изображается прямоугольником со скругленными углами.

Экземпляр зависимой сущности определяется только через отношение к родительской сущности. При установлении идентифицирующей связи атрибуты первичного ключа родительской сущности автоматически переносятся в состав первичного ключа дочерней сущности. Эта операция дополнения атрибутов дочерней сущности при создании связи называется миграцией атрибутов. В дочерней сущности новые атрибуты помечаются как внешние ключи (FK).

При установлении неидентифицирующей связи дочерняя сущность остается независимой,

а атрибуты первичного ключа родительской сущности мигрируют в состав неключевых компонентов дочерней. Неидентифицирующая связь служит для связи независимых сущностей.

Идентифицирующая связь показывается на диаграмме сплошной линией с жирной точкой на дочернем конце связи, неидентифицирующая - пунктирной. Для неидентифицирующей связи можно указать обязательность (Nulls в закладке General диалога Relationship Editor). В случае обязательной связи (No Nulls) при генерации схемы БД атрибут внешнего ключа получит признак NOT NULL, несмотря на то, что внешний ключ не войдет в состав первичного ключа дочерней сущности. В случае необязательной связи (Nulls Allowed) внешний ключ может принимать значение NULL. Необязательная неидентифицирующая связь помечается прозрачным ромбом со стороны родительской сущности

Имя связи (Verb Phrase) - фраза, характеризующая отношение между родительской и дочерней сущностями. Для связи один-ко-многим идентифицирующей или неидентифицирующей достаточно указать имя, характеризующей отношение от родительской к дочерней сущности (Parent-to-Child). Для связи многие-ко-многим следует указывать имена как Parent-to-Child, так и Child-to-Parent. Для отображения имени следует в контекстном меню, которое появляется, если щелкнуть правой кнопкой мыши по любому месту диаграммы, не занятому объектами модели, выбрать пункт Display Options/Relationship и затем включить опцию Verb Phrase.

Имя роли или функциональное имя (Rolename) - это синоним атрибута внешнего ключа, который показывает, какую роль играет атрибут в дочерней сущности. Задать имя роли можно в закладке Rolename/RI Actions диалога Relationship Editor.

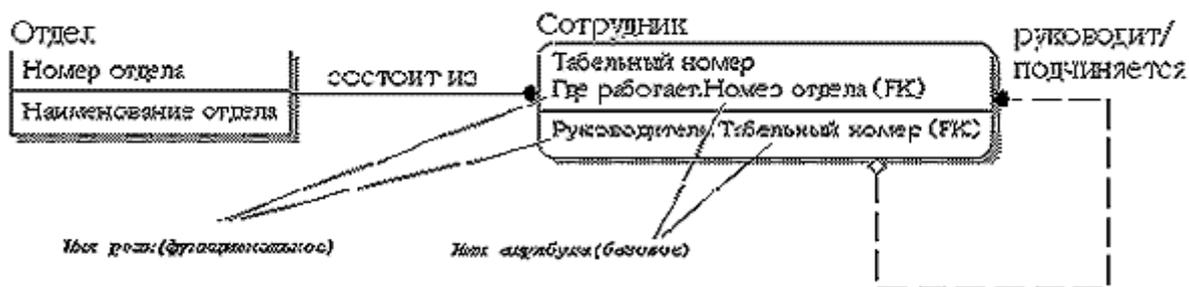


Рис.8. Имена ролей внешних ключей

В примере, приведенном на рис.8, в сущности Сотрудник внешний ключ Номер отдела имеет имя роли "Где работает", которое показывает, какую роль играет этот атрибут в сущности. По умолчанию в списке атрибутов показывается только имя роли. Для отображения полного имени атрибута (как функционального имени, так и имени роли) следует в контекстном меню, которое появляется, если щелкнуть правой кнопкой мыши по любому месту диаграммы, не занятому объектами модели, выбрать пункт Display Options/Entities и затем включить опцию Rolename/Attribute. Полное имя показывается как функциональное имя и базовое имя, разделенные точкой (рис.8). Обязательным является применение имен ролей в том случае, когда два или более атрибутов одной сущности определены по одной и той же области, т.е. они имеют одну и ту же область значений, но разный смысл.

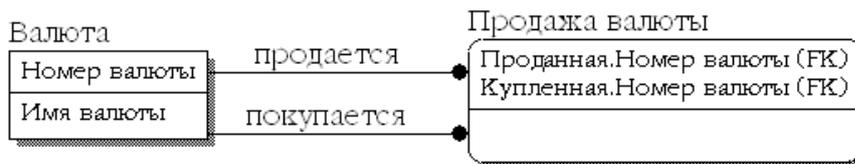


Рис.9. Случай обязательности имен ролей

На рис.9 сущность Продажа валюты содержит информацию об акте обмена валюты, в котором участвуют две валюты - проданная и купленная. Информация о валютах содержится в сущности Валюта. Следовательно, сущности Продажа валюты и Валюта должны быть связаны дважды, и первичный ключ - Номер валюты должен дважды мигрировать в сущность Валюта в качестве внешнего ключа. Необходимо различать эти атрибуты, которые содержат информацию о номере проданной и купленной валюты (имеют разный смысл), но ссылаются на одну и ту же сущность Валюта (имеют общую область значений). В примере на рис.9 атрибуты получили имена ролей Проданная и Купленная.

Другим примером обязательного применения имен ролей являются рекурсивные связи, когда одна и та же сущность является и родительской и дочерней одновременно.

Правила ссылочной целостности (Referential Integrity (RI)) - логические конструкции, которые выражают бизнес-правила использования данных и представляют собой правила вставки, замены и удаления. Задать правила ссылочной целостности можно в закладке Rolename/RI Actions диалога Relationship Editor. При генерации схемы БД на основе опций логической модели будут сгенерированы правила декларативной ссылочной целостности, которые должны быть предписаны для каждой связи, и триггеры, обеспечивающие ссылочную целостность.

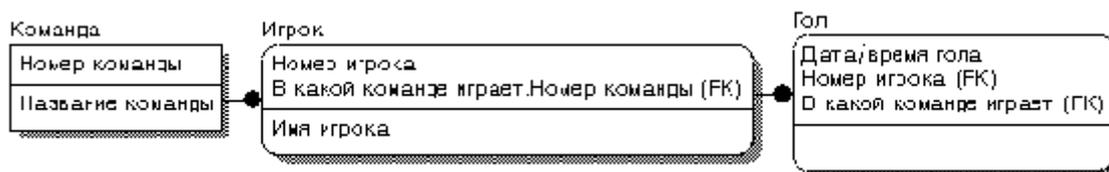


Рис.10. Миграция имен ролей

На рис.10 существует идентифицирующая связь между сущностями Команда и Игрок. Что будет, если удалить команду? Экземпляр сущности Игрок не может существовать без команды (атрибут первичного ключа *В какой команде играет. Номер команды* не может принимать значение NULL), следовательно нужно либо запретить удаление команды, пока в ней числится хотя бы один игрок, либо удалять вместе с командой и всех ее игроков. Такие правила удаления (Parent Delete) называются Parent Restrict (ограничение) и Parent Cascade (каскад). Сущности Игрок и Гол, в свою очередь, тоже связаны идентифицирующей связью и, если на удаление игрока наложено правило каскадного удаления всех записей о его голах, то при удалении команды будут удалены все игроки команды и все голы, забитые этими игроками.

Связь многие-ко-многим возможна только на уровне логической модели данных. Такая связь обозначается сплошной линией с двумя точками на концах. Для внесения связи следует сначала нажать на кнопку  в палитре инструментов (ERwin Toolbox), а затем по очереди щелкнуть по обеим связанным сущностям. Связь многие-ко-многим должна именоваться (Verb Phrase) двумя фразами - в обе стороны. Это облегчает чтение диаграммы.

Создание ключей.

Каждый экземпляр сущности должен быть уникален и отличаться от других атрибутов.

Первичный ключ (primary key) - это атрибут или группа атрибутов, однозначно идентифицирующие экземпляр сущности. Атрибуты первичного ключа на диаграмме не требуют специального обозначения - это те атрибуты, которые находятся в списке атрибутов выше горизонтальной линии. При внесении нового атрибута в диалоге Attribute Editor для того, чтобы сделать его атрибутом первичного ключа, нужно включить флажок Primary Key в нижней части закладки General. На диаграмме ключевой атрибут можно внести в состав первичного ключа, воспользовавшись режимом переноса атрибутов

(кнопка  в палитре инструментов).

В одной сущности может оказаться несколько атрибутов или наборов атрибутов, претендующих на роль первичного ключа. Такие претенденты называются **потенциальными ключами (candidate key)**. Ключи могут быть сложными, т.е. содержащими несколько атрибутов. Сложные первичные ключи не требуют специального обозначения - это список атрибутов выше горизонтальной линии. При выборе первичного ключа предпочтение должно отдаваться более простым ключам, т.е. ключам, содержащим меньшее количество атрибутов. Многие сущности имеют только один потенциальный ключ. Такой ключ становится первичным. Некоторые сущности могут иметь более одного возможного ключа. Тогда один из них становится первичным, а остальные - альтернативными ключами. **Альтернативный ключ (Alternative Key)** - это потенциальный ключ, не ставший первичным.

Каждому ключу соответствует индекс, имя которого также присваивается автоматически. Имена ключа и индекса при желании можно изменить вручную. На диаграмме атрибуты альтернативных ключей обозначаются как (A_n.m.), где n - порядковый номер ключа, m - порядковый номер атрибута в ключе. Когда альтернативный ключ содержит несколько атрибутов, (A_n.m.) ставится после каждого.

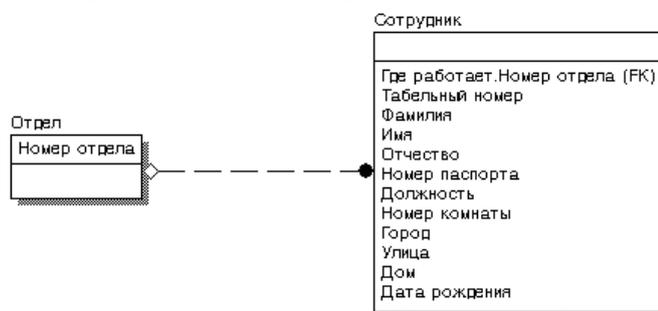


Рис.11. Сущность "Сотрудник" с отображением ключей

Внешние ключи (Foreign Key) создаются автоматически, когда связь соединяет сущности: связи образуют ссылку на атрибуты первичного ключа в дочерней сущности и эти атрибуты образуют внешний ключ в дочерней сущности (миграция ключа). Атрибуты внешнего ключа обозначаются символом (FK) после своего имени (рис.11). Атрибуты внешнего ключа Где работает.Номер отдела ("Где работает" - имя роли) сущности Сотрудник является атрибутом первичного ключа (PK) в сущности Отдел. Зависимая сущность может иметь один и тот же ключ из нескольких родительских сущностей. Сущность может также получить один и тот же внешний ключ несколько раз от одного и того же родителя через несколько разных связей. Когда ERwin обнаруживает одно из этих событий, он распознает, что два атрибута одинаковы, и помещает атрибуты внешнего ключа в зависимой сущности только один раз. Это комбинирование или объединение идентичных атрибутов называется унификацией. Есть случаи, когда унификация нежелательна. Например, когда два атрибута имеют одинаковые имена, но на самом деле они отличаются по смыслу, и необходимо, чтобы это отличие отражалось в диаграмме. В этом случае необходимо использовать имена ролей внешнего ключа (рис.9).

Домены.

Домен можно определить как совокупность значений, из которых берутся значения атрибутов. Каждый атрибут может быть определен только на одном домене, но на каждом домене может быть определено множество атрибутов. В понятие домена входит не только тип данных, но и область значений данных. Например, домен "Возраст" можно определить как положительное целое число и определить атрибут Возраст сотрудника как принадлежащий этому домену. В ERwin домен может быть определен только один раз и использоваться как в логической,

так и в физической модели . На логическом уровне домены можно описать без конкретных физических свойств. На физическом уровне они получают специфические свойства, которые можно изменить вручную. Так, домен "Возраст" может иметь на логическом уровне тип Number, на физическом уровне домену будет присвоен тип INTEGER. Для создания домена в логической модели служит диалог Domain Dictionary Editor. Его можно вызвать из меню Edit/Domain Dictionary по кнопке, расположенной в верхней левой части закладки General диалога Attribute Editor. Для создания нового домена в диалоге Domain Dictionary Editor следует:

- щелкнуть по кнопке New. Появляется диалог New Domain;
- выбрать родительский домен из списка Domain Parent. Новый домен можно создать на основе уже созданного пользователем домена, либо на основе изначально существующего. По умолчанию Erwin имеет четыре предопределенных доменов (String, Number, Blob, Datetime). Новый домен наследует все свойства родительского домена. Эти свойства в дальнейшем можно переопределить;
- набрать имя домена в поле Logical Name. Можно также указать имя домена на физическом уровне в поле Physical Name. Если физическое имя не указано, по умолчанию оно принимает значение логического имени;
- щелкнуть по кнопке ОК;

В диалоге Domain Dictionary Editor можно связать домен с иконкой, с которой он будет отображаться в списке доменов (Domain Icon), иконкой, с которой атрибут, определенный на домене будет отображаться в модели (Icon Inherited by Attribute). Каждый домен может быть описан в закладке Definition, снабжен комментарием в закладке Note или свойством определенным пользователем в закладке UDP. ERwin имеет специальный инструмент, который значительно облегчает создание новых атрибутов в модели, используя описание доменов, - Independent Attribute Browser. Этот диалог вызывается (и скрывается) по горячему ключу CTRL+B. С его помощью можно выбрать в списке домен и по методу drag&drop перенести его в какую-либо сущность. В ней будет создан новый атрибут с именем, которое следует задать в окне Name Inherited by Attribute диалога Domain Dictionary Editor. Если значение поля не задано, по умолчанию принимается имя домена. На физическом уровне диалог Domain Dictionary Editor позволяет редактировать физические свойства домена. Имя этой закладки зависит от выбранного сервера БД. На ней можно задать конкретный тип данных, соответствующих домену, правила присвоения NULL - значений, правила валидации (правила проверки допустимых значений) и задания значения по умолчанию. Правила валидации и значения по умолчанию должны быть предварительно описаны и именованы. Для вызова диалогов редактирования правил валидации и значений по умолчанию служат кнопки  справа от соответствующего списка выбора (Valid и Default).
Функции других закладок диалога Domain Dictionary Editor:
General. Задание родительского домена (Domain Parent) и имени, присваиваемого колонке при ее создании с помощью Independent Column Browser. С помощью опции Physical Only домен можно определить только на уровне физической модели.
Comment. Внесение комментария к атрибуту.
UDP. Свойства, определяемые пользователем.
Visual Basic - PowerBuilder. Задание специальных свойств домена для кодогенерации клиентского приложения.

Задание.

На основе ранее созданной функциональной модели и описания заданного отдела создать логическую модель с использованием пакета ERwin.

Вопросы.

1. Каково назначение пакета ERwin и его основные функции?

2. В чем состоят главные преимущества пакета ERwin?
3. Опишите этапы построения информационной модели.
4. Из каких элементов состоит диаграмма "сущность-связь"?
5. Опишите характеристики связей в методологии IDEF1X.
6. Какие типы ключей используются в пакете ERwin, каково их назначение?
7. Каково предназначение доменов, приведите примеры доменов различного вида.