

Лабораторная работа N2

Задание. Построить модель процессов для графического представления прецедентов, описанных в первой лабораторной работе. Создать и посмотреть разные типы отчетов

Теоретические сведения

1. Методология IDEFO

1.1. Принципы построения модели IDEFO

На начальных этапах создания ИС необходимо понять, как работает организация, которую собираются автоматизировать. Никто в организации не знает, как она работает в той мере подробности, которая необходима для создания ИС. Руководитель хорошо знает работу в целом, но не в состоянии вникнуть в детали работы каждого рядового сотрудника. Рядовой сотрудник хорошо знает, что творится на его рабочем месте, но плохо знает, как работают коллеги. Поэтому для описания работы предприятия необходимо построить модель. Такая модель должна быть адекватна предметной области, следовательно, она должна содержать в себе знания всех участников бизнес-процессов организации.

Наиболее удобным языком моделирования бизнес-процессов является IDEFO, предложенный более 20 лет назад Дугласом Россом (SoftTech, Inc.) и называвшийся первоначально SADT - Structured Analysis and Design Technique. (Подробно методология SADT излагается в книге Дэвида А. Марка и Клемента Мак-Гоуэна "Методология структурного анализа и проектирования АОР - М.:Метатехнология, 1993.) В начале 70-х годов вооруженные силы США применили подмножество SADT, касающееся моделирования процессов, для реализации проектов в рамках программы ICAM (Integrated Computer-Aided Manufacturing). В дальнейшем это подмножество SADT было принято в качестве федерального стандарта США под наименованием IDEFO. Подробные спецификации на стандарты IDEF можно найти на сайте <http://www.idef.com>. В IDEFO 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.

Под моделью в IDEFO понимают описание системы (текстовое и графическое), которое должно дать ответ на некоторые заранее определенные вопросы. Моделируемая система рассматривается как произвольное подмножество Вселенной. Произвольное потому, что, во-первых, мы сами умозрительно определяем, будет ли некий объект компонентом системы, или мы будем его рассматривать как внешнее воздействие, и, во-вторых, оно зависит от точки зрения на систему. Система имеет границу, которая отделяет ее от остальной Вселенной. Взаимодействие системы с окружающим миром описывается как вход (нечто, что перерабатывается системой), выход (результат деятельности системы), управление (стратегии и процедуры, под управлением которых производится работа) и механизм (ресурсы, необходимые для проведения работы). Находясь под управлением, система преобразует входы в выходы, используя механизмы.

Процесс моделирования какой-либо системы в IDEFO начинается с определения контекста, т. е. наиболее абстрактного уровня описания системы в целом. В контекст входит определение субъекта моделирования, цели и точки зрения на модель.

Область моделирования (Scope). Под субъектом понимается сама система, при этом необходимо точно установить, что входит в систему, а что лежит за ее пределами, другими словами, мы должны определить, что мы будем в дальнейшем рассматривать как компоненты системы, а что как внешнее воздействие. На определение субъекта системы будет существенно влиять позиция, с которой рассматривается система, и цель моделирования - вопросы, на которые построенная модель должна дать ответ. Другими словами, первоначально необходимо определить **область** моделирования. Описание области как системы в целом, так и ее компонентов является основой построения модели. Хотя предполагается, что в течение моделирования область может

корректироваться, она должна быть в основном сформулирована изначально, поскольку именно область определяет направление моделирования и когда должна быть закончена модель. При формулировании области необходимо учитывать два компонента - широту и глубину. Широта подразумевает определение границ модели - мы определяем, что будет рассматриваться внутри системы, а что снаружи. Глубина определяет, на каком уровне детализации модель является завершённой. При определении глубины системы необходимо не забывать об ограничениях времени - трудоёмкость построения модели растёт в геометрической прогрессии от глубины декомпозиции. После определения границ модели предполагается, что новые объекты не должны вноситься в моделируемую систему; поскольку все объекты модели взаимосвязаны, внесение нового объекта может быть не просто арифметической добавкой, но в состоянии изменить существующие взаимосвязи. Внесение таких изменений в готовую модель является, как правило, очень трудоёмким процессом (так называемая проблема "плавающей области").

Цель моделирования (Purpose). Модель не может быть построена без четко сформулированной цели. Цель должна отвечать на следующие вопросы:

- Почему этот процесс должен быть промоделирован?
- Что должна показывать модель?
- Что может получить читатель?

Формулировка цели позволяет команде аналитиков сфокусировать усилия в нужном направлении. Примерами формулирования цели могут быть следующие утверждения: "Идентифицировать и определить текущие проблемы, сделать возможным анализ потенциальных улучшений", "Идентифицировать роли и ответственность служащих для написания должностных инструкций", "Описать функциональность предприятия с целью написания спецификаций информационной системы" и т. д.

Точка зрения (Viewpoint). Хотя при построении модели учитываются мнения различных людей, модель должна строиться с единой точки зрения. Точку зрения можно представить как взгляд человека, который видит систему в нужном для моделирования аспекте. Точка зрения должна соответствовать цели моделирования. Очевидно, что описание работы предприятия с точки зрения финансиста и технолога будет выглядеть совершенно по-разному, поэтому в течение моделирования важно оставаться на выбранной точке зрения. Как правило, выбирается точка зрения человека, ответственного за моделируемую работу в целом. Часто при выборе точки зрения на модель важно задокументировать дополнительные альтернативные точки зрения. Для этой цели обычно используют диаграммы *FEO (For Exposition Only)*, которые будут описаны в дальнейшем. IDEFO-модель предполагает наличие четко сформулированной цели, единственного субъекта моделирования и одной точки зрения. Для внесения области, цели и точки зрения в модели IDEFO в VPwin следует выбрать пункт меню **Edit/Model Properties**, вызывающий диалог Model Properties (рис. 1). В закладке Purpose следует внести цель и точку зрения, а в закладку Definition - определение модели и описание области.

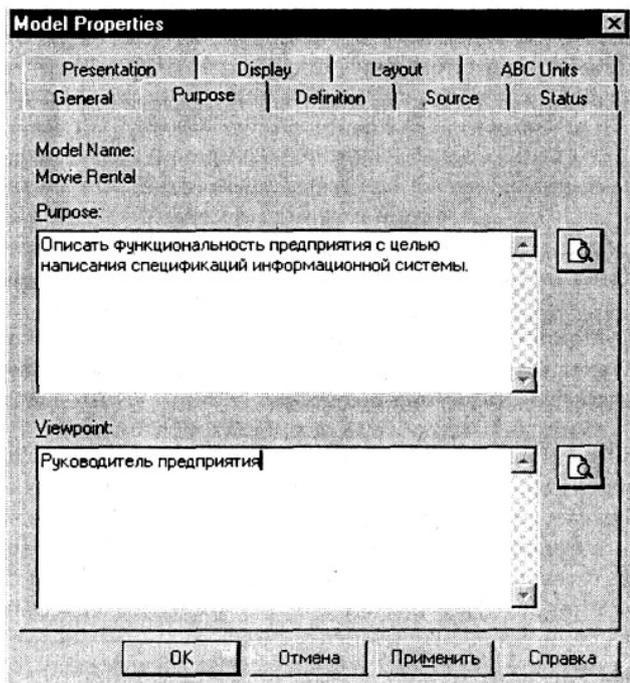


Рис. 1. Диалог задания свойств модели

В закладке Status того же диалога можно описать статус модели (черновой вариант, рабочий, окончательный и т. д.), время создания и последнего редактирования (отслеживается в дальнейшем автоматически по системной дате). В закладке Source описываются источники информации для построения модели (например, "Опрос экспертов предметной области и анализ документации"). Закладка General служит для внесения имени проекта и модели, имени и инициалов автора и временных рамок модели - AS-IS и TO-BE.

Модели AS-IS и TO-BE. Обычно сначала строится модель существующей организации работы - AS-IS (как есть). На основе модели AS-IS достигается консенсус между различными единицами бизнеса по тому, "кто что сделал" и что каждая единица бизнеса добавляет в процесс. Модель AS-IS позволяет выяснить, "что мы делаем сегодня" перед тем, как перепрыгнуть на то, "что мы будем делать завтра". Анализ функциональной модели позволяет понять, где находятся наиболее слабые места, в чем будут состоять преимущества новых бизнес-процессов и насколько глубоким изменениям подвергнется существующая структура организации бизнеса. Детализация бизнес-процессов позволяет выявить недостатки организации даже там, где функциональность на первый взгляд кажется очевидной. Признаками неэффективной деятельности могут быть бесполезные, неуправляемые и дублирующиеся работы, неэффективный документооборот (нужный документ не оказывается в нужном месте в нужное время), отсутствие обратных связей по управлению (на проведение работы не оказывает влияния ее результат), входу (объекты или информация используются нерационально) и т. д. Найденные в модели AS-IS недостатки можно исправить при создании модели TO-BE (как будет) - модели новой организации бизнес-процессов. Модель нужна TO-BE для анализа альтернативных/лучших путей выполнения работы и документирования того, как компания будет делать бизнес в будущем.

Следует указать на распространенную ошибку при создании модели AS-IS - это создание идеализированной модели. Примером может служить создание модели на основе знаний руководителя, а не конкретного исполнителя работ. Руководитель знаком с тем, как предполагается выполнение работы по руководствам и должностным инструкциям и часто не знает, как на самом деле подчиненные выполняют рутинные работы. В результате получается приукрашенная, искаженная модель, которая несет ложную информацию и которую невозможно в дальнейшем использовать для анализа. Такая модель называется 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 TO-BE, и только на основе модели TO-BE

строится модель данных, прототип и затем окончательный вариант ИС. Построение системы на основе модели AS-IS приводит к автоматизации предприятия 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 сопровождение того и другого.

Иногда текущая AS-IS и будущая TO-BE модели различаются очень сильно, так что переход от начального к конечному состоянию становится неочевидным. В этом случае необходима третья модель, описывающая процесс перехода от начального к конечному состояния системы, поскольку такой переход - это тоже бизнес-процесс.

Результат описания модели можно получить в отчете *Model Report*. Диалог настройки отчета по модели вызывается из пункта меню **Report/Model Report**. В диалоге настройки следует выбрать необходимые поля, при этом автоматически отображается очередность вывода информации в отчет (рис. 2).

Model Name: Movie Rental

Definition: Учебная модель

Scope: Технологические, Финансовые и управленческие аспекты деятельности гипотетического предприятия.

Viewpoint: Руководитель предприятия

Purpose: Описать Функциональность предприятия с целью написания спецификаций информационной системы.

Рис. 2. Отчет по модели

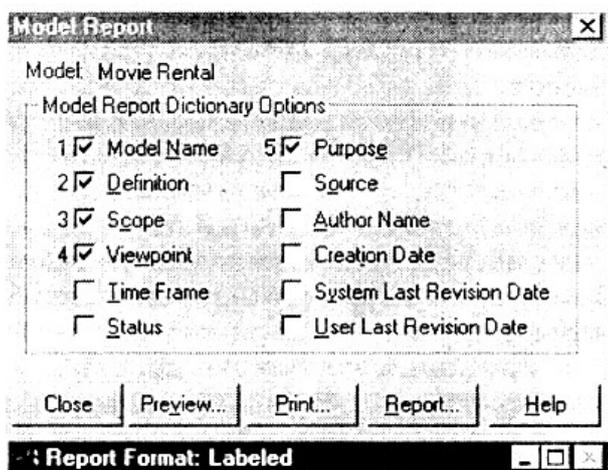


Рис.3. Задание свойств отчета

Диаграммы IDEF0. Основу методологии IDEF0 составляет графический язык описания бизнес-процессов. Модель в нотации IDEF0 представляет собой совокупность иерархически упорядоченных и взаимосвязанных диаграмм. Каждая диаграмма является единицей описания системы и располагается на отдельном листе.

Модель может содержать четыре типа диаграмм:

- контекстную диаграмму (в каждой модели может быть только одна контекстная диаграмма);
- диаграммы декомпозиции;
- 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. 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 проводится декомпозиция каждого большого фрагмента системы на более мелкие и так далее, до достижения нужного уровня подробности описания. После каждого сеанса декомпозиции проводятся сеансы экспертизы - эксперты предметной области 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 много, 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.

1.2. Работы (Activity)

Работы обозначают поименованные процессы, функции или задачи, которые происходят в течение определенного времени и имеют распознаваемые результаты. Работы изображаются в виде прямоугольников. Все работы должны быть названы и определены. Имя работы должно быть выражено отглагольным существительным, обозначающим действие (например, "Изготовление детали", "Прием заказа" и т. д.). Работа "Изготовление детали" может иметь, например, следующее определение: "Работа относится к полному циклу изготовления изделия от контроля качества сырья до отгрузки готового упакованного изделия". При создании новой модели (меню **File/New**) автоматически создается контекстная диаграмма с единственной работой, изображающей систему в целом (рис. 5).

Для внесения имени работы следует щелкнуть по работе правой кнопкой мыши, выбрать в меню Name Editor и в появившемся диалоге внести имя работы. Для описания других свойств работы служит диалог Activity Properties (рис. 4).

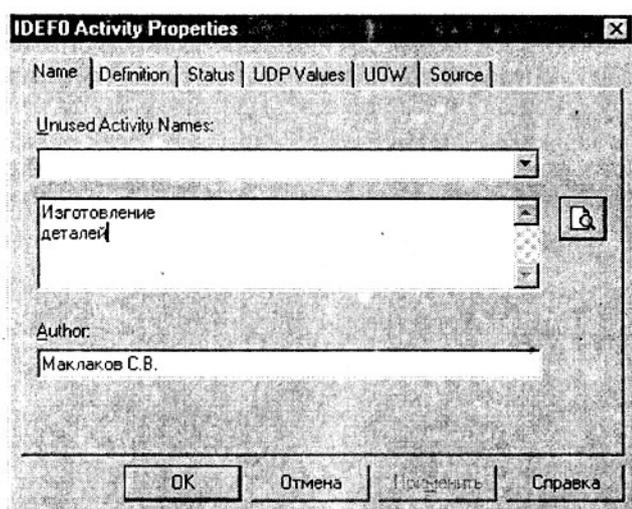


Рис. 4. Редактор задания свойств работы



Рис.5. Пример контекстной диаграммы

Вход (Input) - материал или информация, которые используются или преобразуются работой для получения результата (выхода). Допускается, что работа может не иметь ни одной стрелки входа. Каждый тип стрелок подходит к определенной стороне прямоугольника, изображающего работу, или выходит из нее. Стрелка входа рисуется как входящая в левую грань работы. При описании технологических процессов (для этого и был придуман IDEF0) не возникает проблем определения входов. Действительно, **"Сырье"** на рис. 5 - это нечто, что перерабатывается в процессе "Изготовление изделия" для получения результата. При моделировании ИС, когда стрелками являются не физические объекты, а данные, не все так очевидно. Например, при **"Приеме пациента"** карта пациента может быть и на входе и на выходе, между тем качество этих данных меняется. Другими словами, в нашем примере для того, чтобы оправдать свое назначение, стрелки входа и выхода должны быть точно определены с тем, чтобы указать на то, что данные действительно были переработаны (например, на выходе - **"Заполненная карта пациента"**). Очень часто сложно 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. Стрелка управления рисуется как входящая в верхнюю грань работы. На рис. 5 стрелки **"Задание"** и **"Чертеж"** - управление для работы **"Изготовление изделия"**. Управление влияет на работу, но не преобразуется работой. Если цель работы - изменить процедуру или стратегию, то такая процедура или стратегия будет для работы входом. В случае возникновения неопределенности в статусе стрелки (управление или вход) рекомендуется рисовать стрелку управления.

Выход (Output) - материал или информация, которые производятся работой. Каждая работа должна иметь хотя бы одну стрелку выхода. Работа без результата не имеет смысла и не должна моделироваться. Стрелка выхода рисуется как исходящая из правой грани работы. На рис. 1.5 стрелка **"Готовое изделие"** является выходом для работы **"Изготовление изделия"**.

Механизм (Mechanism) - ресурсы, которые выполняют работу, например персонал предприятия, станки, устройства и т. д. Стрелка механизма рисуется как входящая в нижнюю грань работы. На рис. 5 стрелка **"Персонал предприятия"** является механизмом для работы **"Изготовление изделия"**. По усмотрению аналитика стрелки механизма могут не изображаться в модели.

Вызов (Call) - специальная стрелка, указывающая на другую модель работы. Стрелка вызова рисуется как исходящая из нижней грани работы. На рис. 1.5 стрелка **"Другая модель работы"** является вызовом для работы **"Изготовление изделия"**. Стрелка вызова используется для указания того, что некоторая работа выполняется за пределами моделируемой системы.

В VPwin стрелки вызова используются в механизме слияния и разделения моделей.

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 входа следует:

- щелкнуть по кнопке с символом стрелки  в палитре инструментов 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 Name Editor и добавить имя стрелки в закладке Name диалога IDEFO Arrow Properties.

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

Имена вновь внесенных стрелок автоматически заносятся в словарь (Arrow Dictionary).

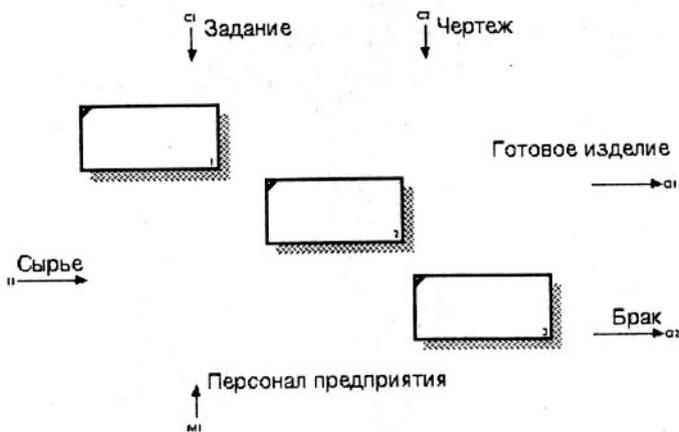


Рис. 6. Пример несвязанных стрелок

На рис. 6 приведен фрагмент диаграммы декомпозиции с несвязанными стрелками, генерирующийся VPwin при декомпозиции работы **"Изготовление изделия"** (см. рис. 5). Для связывания стрелок входа, управления или механизма необходимо перейти в режим редактирования стрелок, щелкнуть по наконечнику стрелки и щелкнуть по соответствующему сегменту работы. Для связывания стрелки выхода необходимо перейти в режим редактирования стрелок, щелкнуть по сегменту выхода работы и затем по стрелке.

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 IDEFO различают пять типов связей работ.

Связь по входу (output-input), когда стрелка выхода вышестоящей работы (далее - просто выход) направляется на вход нижестоящей (например, на рис. 7 стрелка **"Детали"** связывает работы **"Изготовление деталей"** и **"Сборка изделия"**).

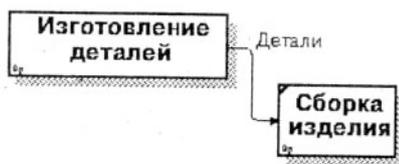


Рис. 7. Связь по входу

Связь по управлению (output-control), когда выход вышестоящей работы направляется на

управление нижестоящей. Связь по управлению показывает доминирование вышестоящей работы. Данные или объекты выхода вышестоящей работы не меняются в нижестоящей. На рис. 8 стрелка "Чертеж" связывает работы "Создание чертежа детали" и "Изготовление детали"), при этом чертеж не претерпевает изменений в процессе изготовления деталей.

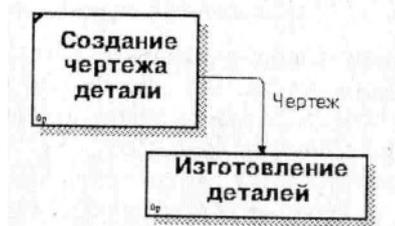


Рис. 8. Связь по управлению

Обратная связь по входу (output-input feedback), когда выход нижестоящей работы направляется на вход вышестоящей. Такая связь, как правило, используется для описания циклов. На рис. 9 стрелка "Брак" связывает работы "Переработка сырья" и "Контроль качества", при этом выявленный на контроле брак направляется на вторичную переработку.

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

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

Несвязанные граничные стрелки (unconnected border arrow). При декомпозиции работы входящие в нее и исходящие из нее стрелки (кроме стрелки вызова) автоматически появляются на диаграмме декомпозиции (миграция стрелок), но при этом не касаются работ. Такие стрелки называются несвязанными и воспринимаются в VPwin как синтаксическая ошибка.

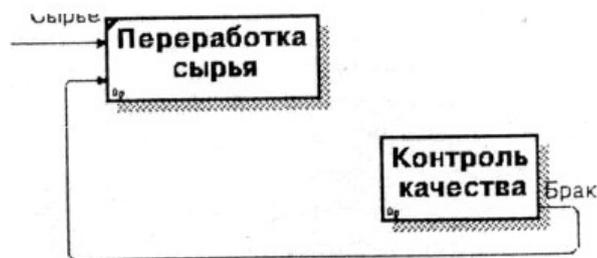


Рис. 9. Обратная связь по входу

Обратная связь по управлению (output-control feedback), когда выход нижестоящей работы направляется на управление вышестоящей (стрелка "Рекомендации", рис. 10). Обратная связь по управлению часто свидетельствует об эффективности бизнес - процесса. На рис. 10 качество изделия может быть повышено путем непосредственного регулирования процессам» изготовления деталей и сборки изделия в зависимости от результат (выхода) работы "Контроль качества".

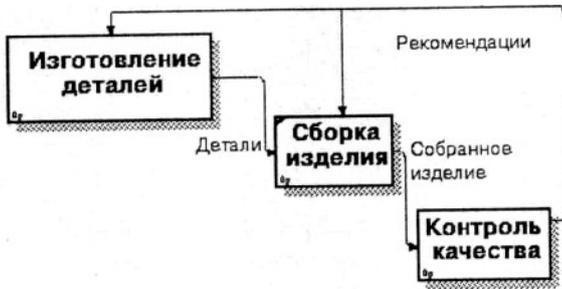


Рис. 10. Обратная связь по управлению

Связь выход-механизм (output-mechanism), когда выход одной работы направляется на механизм другой. Эта взаимосвязь используется реже остальных и показывает, что одна работа подготавливает ресурсы, необходимые для проведения другой работы (рис. 11).

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



Рис. 11. Связь выход-механизм

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 в IDEF0 используются разветвляющиеся и сливающиеся стрелки. Для разветвления стрелки нужно в режиме редактирования стрелки щелкнуть по фрагменту стрелки и по соответствующему сегменту работы. Для слияния двух стрелок выхода нужно в режиме редактирования стрелки сначала щелкнуть по сегменту выхода работы, а затем по соответствующему фрагменту стрелки.

Смысл разветвляющихся и сливающихся стрелок передается именованием каждой ветви стрелок. Существуют определенные правила именования таких стрелок. Рассмотрим их на примере разветвляющихся стрелок. Если стрелка именована до разветвления, а после разветвления ни одна из ветвей не именована, то подразумевается, что каждая ветвь моделирует те же данные или объекты, что и ветвь до разветвления (рис. 12).



Рис. 12. Пример именования разветвляющейся стрелки

Если стрелка именована до разветвления, а после разветвления какая-либо из ветвей не

именована, то подразумевается, что эти ветви соответствуют именованию. Если при этом какая-либо ветвь после разветвления осталась неименованной, то подразумевается, что она моделирует те же данные или объекты, что и ветвь до разветвления (рис. 13).



Рис. 13. Другой пример именованной разветвляющейся стрелки

Недопустима ситуация, когда стрелка до разветвления не именована, а после разветвления не именована какая-либо из ветвей. ВРwin определяет такую стрелку как синтаксическую ошибку (рис. 14).



Рис. 14. Пример неверного именованной разветвляющейся стрелки

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

Тоннелирование стрелок. Вновь внесенные граничные стрелки на диаграмме декомпозиции нижнего уровня изображаются в квадратных скобках и автоматически не появляются на диаграмме верхнего уровня (рис. 15)

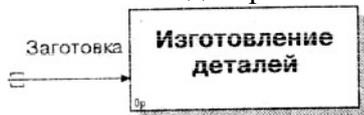


Рис. 15. Неразрешенная (unresolved) стрелка

Для их "перетаскивания" вверх нужно сначала выбрать кнопку



на палитре инструментов и щелкнуть по квадратным скобкам **граничной** стрелки. Появляется диалог Border Arrow Editor (рис. 16).

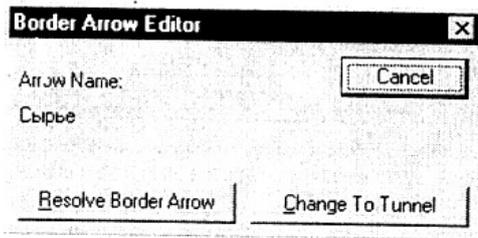


Рис. 16. Диалог Border Arrow Editor

Если щелкнуть по кнопке Resolve Border Arrow, стрелка мигрирует на диаграмму верхнего уровня, если по кнопке Change To Tunnel - стрелка будет затоннелирована и не попадет на другую диаграмму. Тоннельная стрелка изображается с круглыми скобками на конце (рис. 18).

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 (на родительскую диаграмму). Если эти данные не используются на родительской диаграмме, их нужно направить еще выше, и т. д. В результате малозначимая стрелка будет изображена на всех уровнях и затруднит чтение всех диаграмм, на которых она присутствует. Выходом является 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. (Предполагается, что не нужно детализировать стрелку механизма, т. е. стрелка механизма на дочерней работе именована до разветвления, а после разветвления ветви не имеют собственного имени). В этом случае стрелка механизма на нижнем уровне может быть удалена, после чего на водителской диаграмме она может быть затоннелирована, а в комментарии к стрелке или в словаре можно указать, что механизм будет использоваться во всех работах дочерней диаграммы декомпозиции. Такое тоннелирование называется "не-в-дочерней-работе" (рис. 18).



Рис. 18. Типы тоннелирования стрелок

1.4. Нумерация работ и диаграмм

Все работы модели нумеруются. Номер состоит из префикса и числа. Может быть использован префикс любой длины, но обычно использую* префикс А. Контекстная (корневая) работа дерева имеет номер А0. Работы декомпозиции А0 имеют номера А1, А2, А3 и т. д. Работы декомпозиции нижнего уровня имеют номер родительской работы и очередной порядке вый номер, например работы декомпозиции А3 будут иметь номера А31 А32, А33, А34 и т. д. Работы

образуют иерархию, где каждая работа может иметь одну родительскую и несколько дочерних работ, образуя дерево. Та кое дерево называют деревом узлов, а вышеописанную нумерацию - нумерацией по узлам. Имеются незначительные варианты нумерации, которые можно настроить в закладке **Presentation** диалога **Model Properties** (меню **Edit/Model Properties**).

Диаграммы IDEF0 имеют двойную нумерацию. Во-первых, диаграмм имеют номера по узлу. Контекстная диаграмма всегда имеет номер A-1 декомпозиция контекстной диаграммы - номер A0, остальные диаграмм декомпозиции - номера по соответствующему узлу (например, A1, A2, A2, A13 и т. д.). VPwin автоматически поддерживает нумерацию по узлам, т. при проведении декомпозиции создается новая диаграмма и ей автоматически присваивается соответствующий номер. В результате проведения экспертизы диаграммы могут уточняться и изменяться, следовательно, могут быть созданы различные версии одной и той же (с точки зрения ее расположения в дереве узлов) диаграммы декомпозиции. VPwin позволяет иметь в модели только одну диаграмму декомпозиции в данном узле. Превью версии диаграммы можно хранить в виде бумажной копии, либо как FEO диаграмму. (К сожалению, при создании FEO-диаграмм отсутствует возможность отката, т. е. можно получить из диаграммы декомпозиции FES но не наоборот.) В любом случае следует отличать различные версии одной и той же диаграммы. Для этого существует специальный номер - C-number который должен присваиваться автором модели вручную. C-number - произвольная строка, но рекомендуется придерживаться стандарта, который, например состоит из буквенного префикса и порядкового номера, причем в качестве префикса используются инициалы автора диаграммы, а порядковый номер отслеживается автором вручную, например MCB00021.

1.5. Диаграммы дерева узлов и FEO

Диаграмма дерева узлов показывает иерархию работ в модели и позволяет рассмотреть всю модель целиком, но не показывает взаимосвязи между работами (стрелки) (рис. 19). Процесс создания модели работ является итерационным, следовательно, работы могут менять свое расположение в дереве узлов многократно. Чтобы не запутаться и проверить способ декомпозиции, следует после каждого изменения создавать диаграмму дерева узлов. Впрочем, VPwin имеет мощный инструмент навигации по модели - **Model Explorer**, который позволяет представить иерархию работ и диаграмм в удобном и компактном виде, однако этот инструмент не является составляющей стандарта IDEF0.

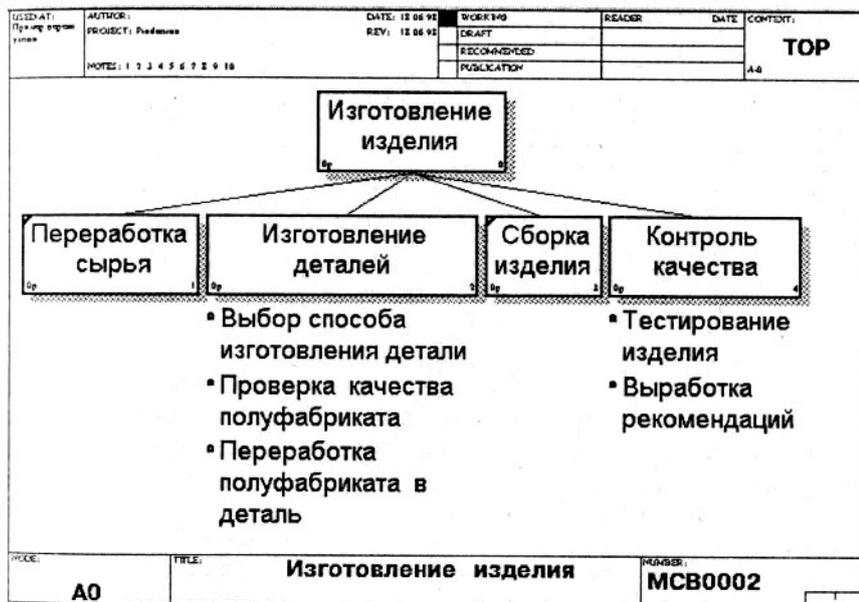


Рис. 19. Диаграмма дерева узлов

Для создания диаграммы дерева узлов следует выбрать в меню пункт **Insert/Node Tree**. Возникает диалог формирования диаграммы дерева узлов **Node Tree Definition**.

Каркас содержит заголовок (верхняя часть рамки) и подвал (нижняя часть). Заголовок каркаса используется для отслеживания диаграммы процессе моделирования. Нижняя часть используется для идентификации позиционирования в иерархии диаграммы.

Смысл элементов каркаса приведен в табл. 1.2 и 1.3.

Таблица 1.2. Поля заголовка каркаса (слева направо)

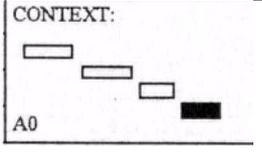
| Поле | Смысл |
|---------------------------|---|
| Used At | Используется для указания на родительскую работу в случае, если на текущую диаграмму ссылались посредством стрелки вызова |
| Autor, Date, Rev, Project | Имя создателя диаграммы, дата создания и имя проекта, в рамках которого была создана диаграмма. REV-дата последнего редактирования диаграммы |
| Notes 12345678910 | 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 | <p>Схема расположения работ в диаграмме верхнего уровня. Работа, являющаяся родительской, показана темным прямоугольником, остальные - светлым. На контекстной диаграмме (A-0) показывается надпись TOP. В левом нижнем углу показывается номер по узлу родительской диаграммы:</p>  |

Таблица 1.3. Поля подвала каркаса (слева направо)

| 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 |
|------|--|

Значения полей каркаса задаются в диалоге Diagram Properties (меню Edit/Diagram Properties) - рис. 21.

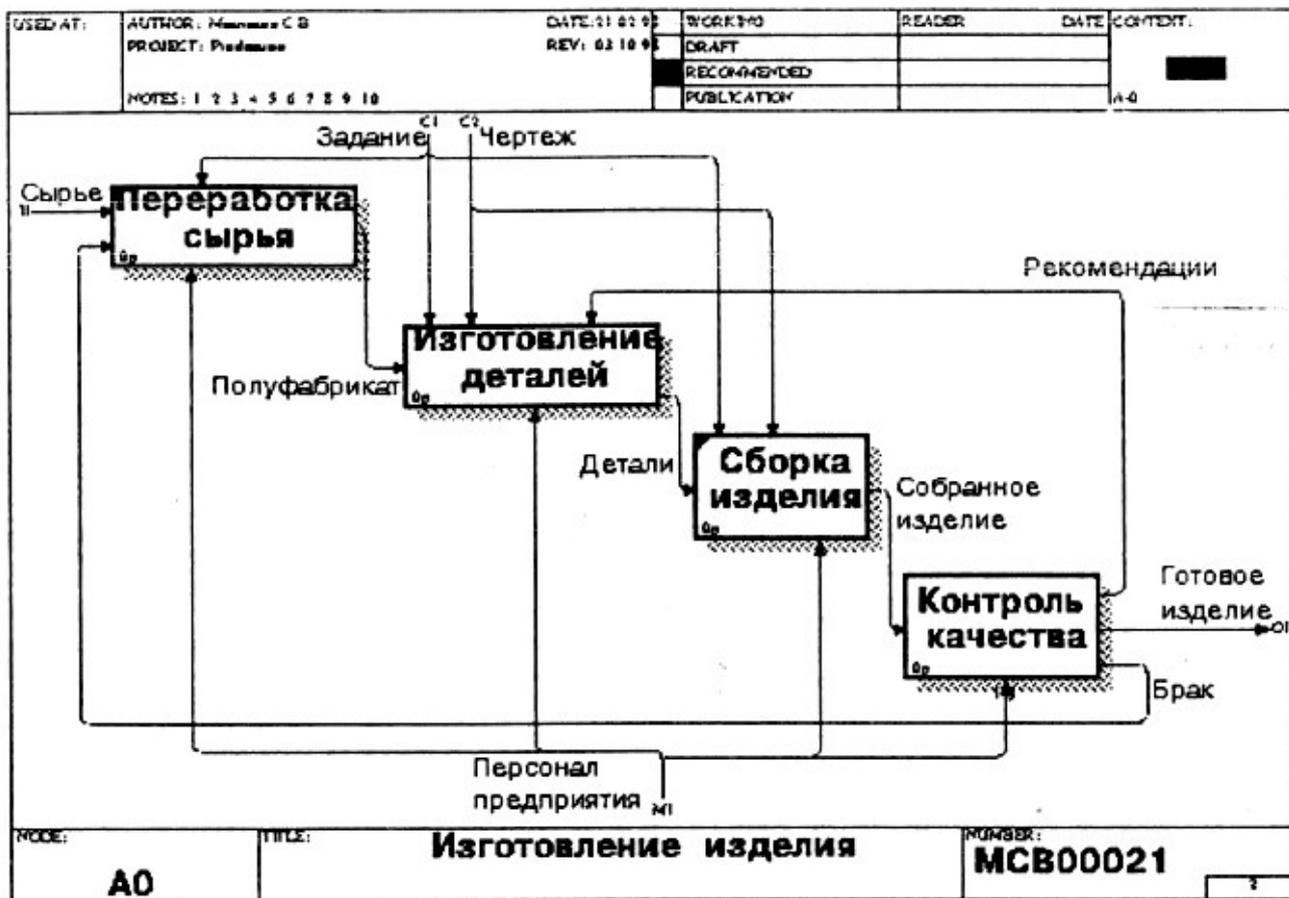


Рис. 21. Диалог Diagram Properties

Лабораторная работа N3

Задание. Продолжить построение модели для графического представления прецедентов, описанных в первой лабораторной работе. Выполнить слияние моделей двух прецедентов. Внести замечания в диаграмму.

Теоретические сведения

1.2.7. Слияние и расщепление моделей

Возможность слияния и расщепления моделей обеспечивает коллективную работу над проектом. Так, руководитель проекта может создать декомпозицию верхнего уровня и дать задание аналитикам продолжить декомпозицию каждой ветви дерева в виде отдельных моделей. После окончания работы над отдельными ветвями все подмодели могут быть слиты в единую модель. С другой стороны, отдельная ветвь модели может быть отщеплена для использования в качестве независимой модели, для доработки или архивирования.

VRwin использует для слияния и разветвления моделей стрелки вызова.

Для слияния необходимо выполнить следующие условия:

- Обе сливаемые модели должны быть открыты в VRwin.
- Имя модели-источника, которое присоединяют к модели-цели, должно совпадать с именем стрелки вызова работы в модели-цели (рис. 22).
- Стрелка вызова должна исходить из недекомпозируемой работы (работа должна иметь диагональную черту в левом верхнем углу) (рис. 23).

Имена контекстной работы подсоединяемой модели-источника и работы на модели-цели, к которой мы подсоединяем модель-источник, должны совпадать (рис. 22).

Модель-источник должна иметь по крайней мере одну диаграмму декомпозиции.

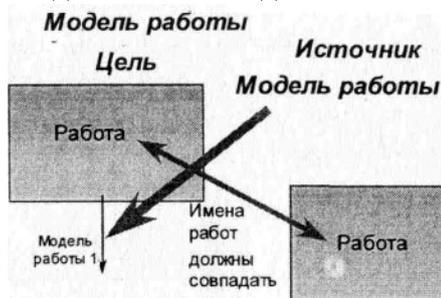


Рис. 22. Условия слияния моделей

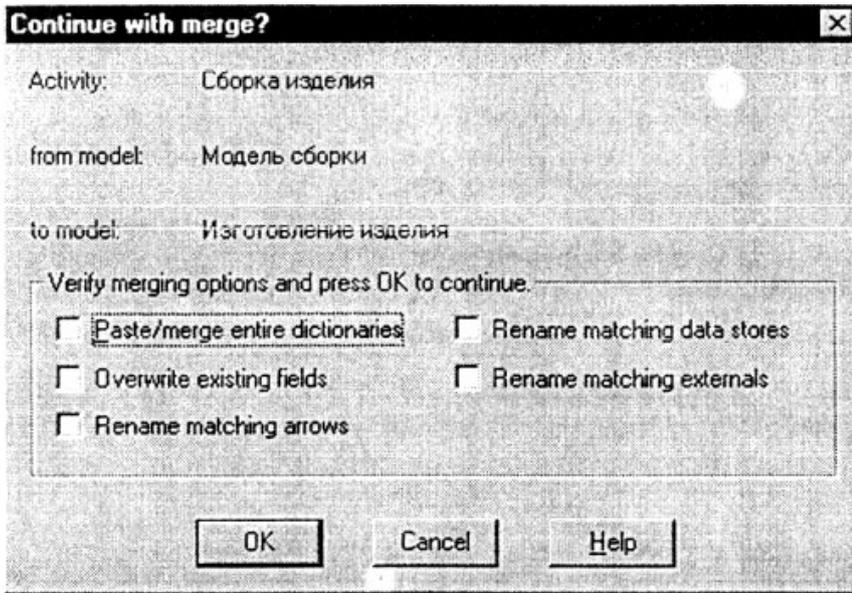
Для слияния моделей нужно щелкнуть правой кнопкой мыши по работе со стрелкой вызова в модели-цели и во всплывающем меню выбрать пункт Merge Model.



Рис. 23. Стрелка вызова работы "Сборка изделия" модели-цели

Появляется диалог, в котором следует указать опции слияния модели (рис. 24). При слиянии моделей объединяются и словари стрелок и работ. В случае одинаковых определений возможна

перезапись определений или принятие определений из модели-источника. То же относится к именам стрелок, хранилищам данных и внешним ссылкам. (Хранилища данных и внешние ссылки - объекты диаграмм потоков данных, DFD, будут рассмотрены ниже.)



7

Рис. 24. Диалог *Continue with merge?*

После подтверждения слияния (кнопка ОК) модель-источник подсоединяется к модели-цели, стрелка вызова исчезает, а работа, от которой отходила стрелка вызова, становится декомпозируемой - к ней подсоединяется диаграмма декомпозиции первого уровня модели-источника. Стрелки, касающиеся работы на диаграмме модели-цели, автоматически не мигрируют в декомпозицию, а отображаются как неразрешенные. Их следует тоннелировать вручную. На рис. 25 показано, как выглядят модели в окне Model Explorer после слияния.

В процессе слияния модель-источник остается неизменной и к модели-цели подключается фактически ее копия. Не нужно путать слияние моделей с синхронизацией. Если в дальнейшем модель-источник будет редактироваться, эти изменения автоматически не попадут в соответствующую ветвь модели-цели.

Разделение моделей производится аналогично. Для отщепления ветви от модели следует щелкнуть правой кнопкой мыши по декомпозированной работе (работа не должна иметь диагональной черты в левом верхнем углу) и выбрать во всплывающем меню пункт Split Model. В появившемся диалоге Split Options следует указать имя создаваемой модели. После подтверждения расщепления в старой модели работа станет недекомпозированной (признак - диагональная черта в левом верхнем углу), будет создана стрелка вызова, причем ее имя будет совпадать с именем новой модели, и, наконец, будет создана новая модель, причем имя контекстной работы будет совпадать с именем работы, от которой была "оторвана" декомпозиция.

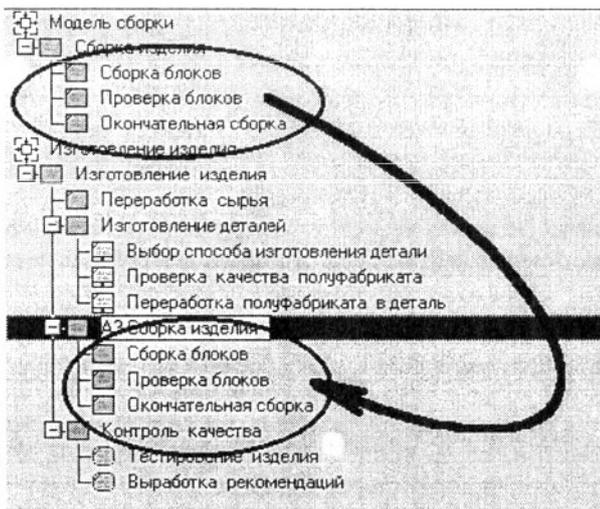


Рис. 25. Модели после слияния

1.8. Рекомендации по рисованию диаграмм

В реальных диаграммах к каждой работе может подходить и от каждой может отходить около десятка стрелок. Если диаграмма содержит 6-8 работ, то она может содержать 30-40 стрелок, причем они могут сливаться, разветвляться и пресекаться. Такие диаграммы могут стать очень плохо читаемыми. В IDEF0 существуют соглашения по рисованию диаграмм, которые призваны облегчить чтение и экспертизу модели. Некоторые из этих правил VPwin поддерживает автоматически, выполнение других следует обеспечить вручную.

- Прямоугольники работ должны располагаться по диагонали с левого верхнего в правый нижний угол (порядок доминирования). При создании новой диаграммы декомпозиции VPwin автоматически располагает работы именно в таком порядке. В дальнейшем можно добавить новые работы или изменить расположение существующих, но нарушать диагональное расположение работ по возможности не следует. Порядок доминирования подчеркивает взаимосвязь работ, позволяет минимизировать изгибы и пересечения стрелок.

- Следует максимально увеличивать расстояние между входящими или выходящими стрелками на одной грани работы. Если включить опцию Line Drawing: Automatically space arrows на закладке Layout диалога Model Properties (меню Edit/Model Properties), VPwin будет располагать стрелки нужным образом автоматически.

Следует максимально увеличить расстояние между работами, поворотами и пересечениями стрелок. Если две стрелки проходят параллельно (начинаются из одной и той же грани одной работы и заканчиваются на одной и той же грани другой работы), то по возможности следует их объединить и назвать единым термином.

Обратные связи по входу рисуются "нижней" петлей, обратная связь по управлению - "верхней". VPwin автоматически рисует обратные связи нужным образом. Его можно "обмануть", но лучше этого не делать.

Циклические обратные связи следует рисовать только в случае крайней необходимости, когда подчеркивают значение повторно используемого объекта. Принято изображать такие связи на диаграмме декомпозиции. VPwin не позволяет создать циклическую обратную связь за один прием. Если все же необходимо изобразить такую связь, следует сначала создать обычную связь по входу, затем разветвить стрелку, направить новую ветвь обратно ко входу работы-источника и, наконец, удалить старую ветвь стрелки выхода (рис. 26).



Рис. 26. Пример обратной циклической связи

Следует минимизировать число пересечений, петель и поворотов стрелок. Это ручная и, в случае насыщенных диаграмм, творческая работа (рис. 27).

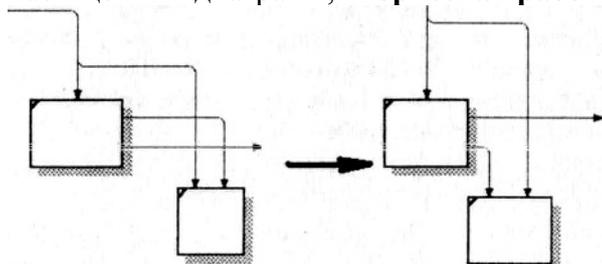


Рис. 27. Минимизация пересечений и поворотов стрелок

Если нужно изобразить связь по входу, необходимо избегать "нависания" работ друг над другом. В этом случае VPwin изображает связи по входу в виде петли, что затрудняет чтение диаграмм (рис. 28).

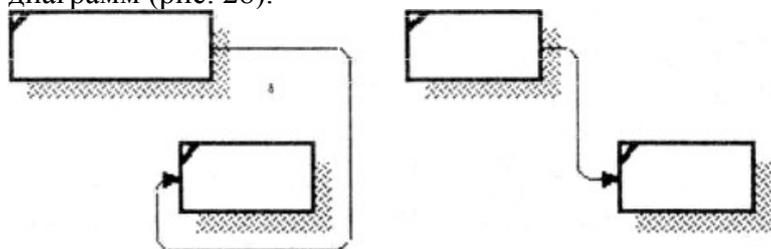


Рис. 28. Пример правильного (справа) и неправильного (слева) расположения работ при изображении связи по входу

1.9. Проведение экспертизы

Цикл автор-читатель. Цикл автор-читатель предназначен для обеспечения обратной связи при построении модели. Он включает определенные формализованные процедуры, предписывающие правила координации деятельности участников создания модели. В работе над моделью принимают участие специалисты разных специальностей - аналитики (авторы), эксперты предметной области (читатели), библиотекари и комитет технического контроля. Обычно библиотекарь выделяется для больших проектов. Цикл автор-читатель содержит следующие этапы:

На очередном этапе декомпозиции аналитик создает диаграмму на основе общих знаний, анализа документации и опроса экспертов. Общие знания не позволяют создать диаграмму достаточно корректно, поэтому она нуждается в уточнении и дополнении.

Все коммуникации при создании модели контролируются библиотекарем. Он ответственен за прохождение папок и архивирование диаграмм модели. После создания диаграмма посылается библиотекарю для помещения в архив.

Автором формируется папка и передается для распространения библиотекарю (одна копия направляется автору). В папку должна входить текущая диаграмма. Кроме того, в папку могут включаться сопутствующие отчеты, в том числе словарь стрелок и работ, диаграмма верхнего уровня, дерево узлов и любая необходимая дополнительная документация. На папке регистрируются входящие данные - дата, автор, данные читателя и т. д., после чего папка направляется эксперту предметной области (читателю).

Читатель рецензирует папку и записывает свои комментарии. Замечания вносятся в диаграмму по определенным правилам. Если читатель решил внести замечание, он должен указать номер замечания, затем внести текст замечания и в каркасе диаграммы в разделе Notes зачеркнуть цифру, соответствующую номеру замечания (рис. 29).

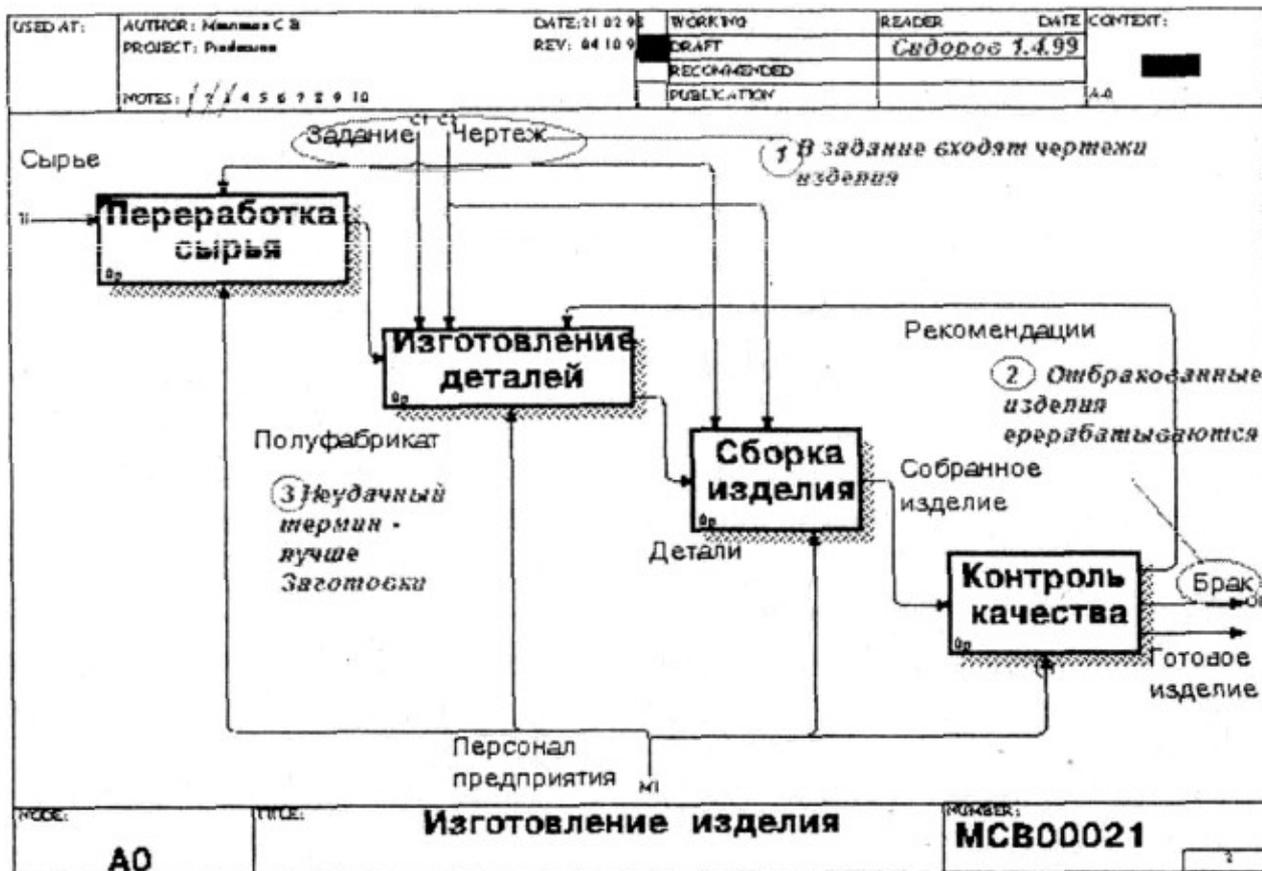


Рис. 29. Внесение замечаний в диаграмму

После рецензирования папки возвращаются библиотекарю. Библиотекарь должен обеспечивать проведение рецензирования в срок. Затем папки регистрируются и направляются автору. Автор вносит ответ на замечания и, если он согласен с замечаниями, вносит изменения в модель. На практике зачастую сеанс экспертизы проводится в форме устного собеседования между автором и экспертом. В этом случае особенно важно вносить замечания эксперта и комментарии автора в диаграмму для документирования всех идей, возникших в результате моделирования.

Если это необходимо, проводится дополнительная экспертиза у того же или у другого эксперта. После прохождения нескольких циклов число замечаний обычно уменьшается и диаграмма становится стабильной. В процессе изменения диаграмма может менять свой статус, который должен быть отражен в каркасе (см. табл. 1.2). Когда автор считает, что диаграмма уже достаточно проработана и достигла уровня "Recommended", он пересылает ее на утверждение в комитет технического контроля, где она проходит окончательную экспертизу. После внесения замечаний и окончательных изменений диаграмма (или набор диаграмм) окончательно утверждается, получает статус "Publication" и может быть распечатана и распространена среди участников проекта.

2. Создание отчетов в BPwin

BPwin имеет мощный инструмент генерации. Отчеты по модели вызываются из пункта меню Report. Всего имеется семь типов отчетов;

1. **Model Report.** Этот отчет уже упоминался в 1.1. Он включает информацию о контексте модели - имя модели, точку зрения, область, цель, имя автора, дату создания и др.
2. **Diagram Report.** Отчет по конкретной диаграмме. Включает список объектов (работ, стрелок, хранилищ данных, внешних ссылок и т. д.).
3. **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.

4. **Activity Cost Report.** Отчет о результатах стоимостного анализа. Будет рассмотрен ниже.

5. **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.

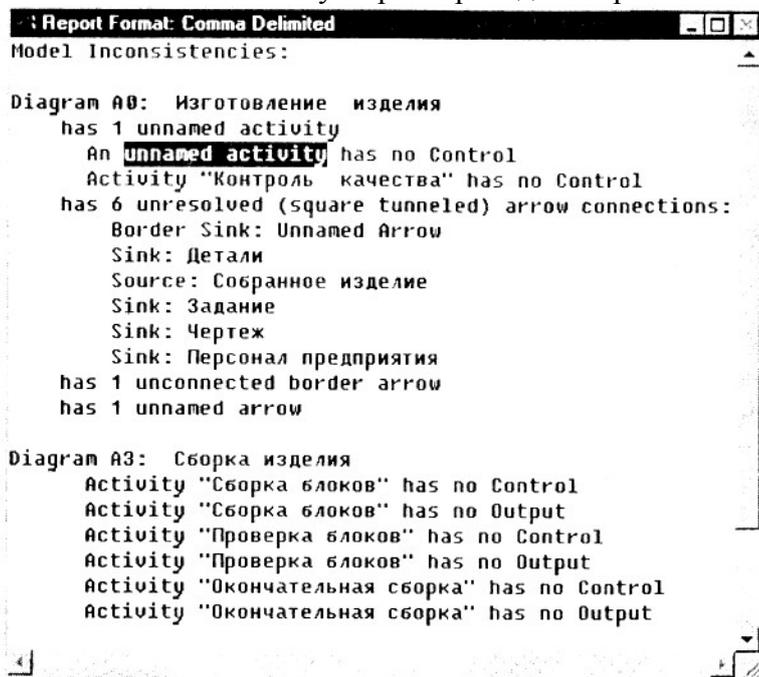
6. **DataUsage Report.** Отчет о результатах связывания модели процессов и модели данных. (Будет рассмотрен ниже.)

7. **Model Consistency Report.** Отчет, содержащий список синтаксических ошибок модели.

Синтаксические ошибки IDEF0 с точки зрения VPwin разделяются на три типа:

- Во-первых, это ошибки, которые VPwin выявить не в состоянии. Например, синтаксис IDEF0 требует, чтобы имя работы было выражено отглагольным существительным или глагольной формой, выражающей действие ("**Изготовление изделия**", "**Обслуживание клиента**", "**Выпуска счета**" и т. д.), а имя стрелки также должно быть выражено существительным. VPwin не позволяет анализировать синтаксис естественного языка (английского и русского) и смысл имен объектов и поэтому игнорирует ошибки этого типа. Выявление таких ошибок - ручная работа, которая ложится на плечи аналитиков и должна контролироваться руководителем проекта.
- Ошибки второго типа VPwin просто не допускает. Например, каждая грань работы предназначена для определенного типа стрелок. VPwin просто не позволит создать на диаграмме IDEF0 внутреннюю стрелку, выходящую из левой грани работы и входящую в правую грань.
- Третий тип ошибок VPwin позволяет допустить, но детектирует их. Полный их список можно получить в отчете **Model Consistency Report**.

Это единственный неопциональный отчет в VPwin. Список ошибок может содержать, например, неименованные работы и стрелки (unnamed arrow, unnamed activity), несвязанные стрелки (unconnected border arrow), неразрешенные стрелки (unresolved (square tunneled) arrow connections), работы, не имеющие по крайней мере одной стрелки выхода и одной стрелки управления (Activity "Сборка блоков" has no Control, Activity "Сборка блоков" has no Output), и т. д. Пример отчета Model Consistency Report приведен на рис. 30.



```
Report Format: Comma Delimited
Model Inconsistencies:

Diagram A0: Изготовление изделия
  has 1 unnamed activity
  An unnamed activity has no Control
  Activity "Контроль качества" has no Control
  has 6 unresolved (square tunneled) arrow connections:
  Border Sink: Unnamed Arrow
  Sink: Детали
  Source: Собранное изделие
  Sink: Задание
  Sink: Чертеж
  Sink: Персонал предприятия
  has 1 unconnected border arrow
  has 1 unnamed arrow

Diagram A3: Сборка изделия
  Activity "Сборка блоков" has no Control
  Activity "Сборка блоков" has no Output
  Activity "Проверка блоков" has no Control
  Activity "Проверка блоков" has no Output
  Activity "Окончательная сборка" has no Control
  Activity "Окончательная сборка" has no Output
```

Рис. 30. Отчет Model Consistency Report

При выборе пункта меню, который соответствует какому-либо отчету, появляется диалог настройки отчета. Для каждого из семи типов отчетов он выглядит по-своему. Рассмотрим типичный диалог Arrow Report (рис. 31).

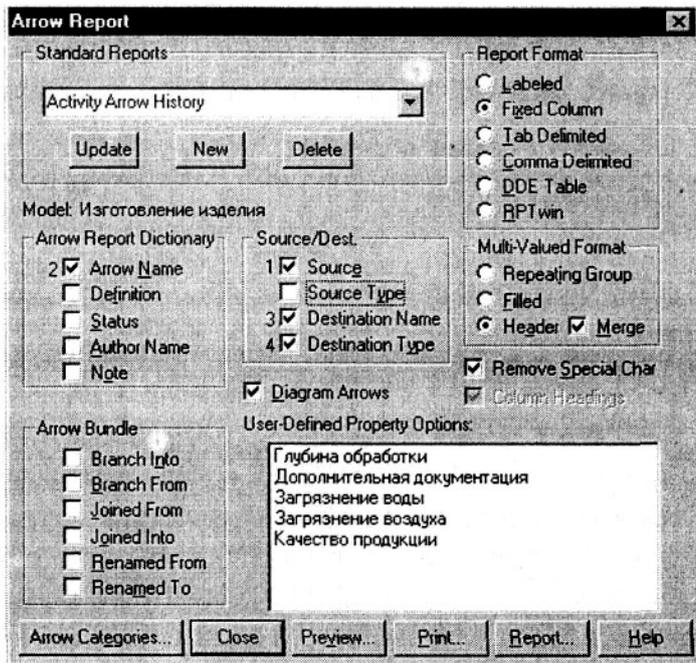


Рис. 31. Диалог Arrow Report

Раскрывающийся список Standard Reports позволяет выбрать один из стандартных отчетов. Стандартный отчет - это запоминаемая комбинация переключателей, флажков и других элементов управления диалога. Для создания собственного стандартного отчета необходимо задать опции отчета, ввести имя отчета в поле списка выбора и щелкнуть по кнопке New. BPwin сохраняет информацию о стандартном отчете в файле BPWINRPT.INI. Все определения этого файла доступны из любой модели. Единственное ограничение - свойства, определяемые пользователем (User-Defined Properties). Они сохраняются в виде указателя и поэтому доступны только из "родной" модели. Стандартный отчет можно изменить (кнопка Update) или удалить (кнопка Delete).

В правом верхнем углу диалога находится группа управляющих элементов для выбора формата отчета. Доступны следующие форматы:

- Labeled - отчеты включают метку поля, затем, в следующей строке, печатается содержимое поля;
- Fixed Column - каждое поле печатается в собственной колонке;
- Tab-Comma Delimited - каждое поле печатается в собственной колонке. Колонки разделяются знаком табуляции или запятыми;
- DDE Table - данные передаются по DDE приложению, например MS Word или Excel;
- RPTwin - отчет создается в формате Platinum RPTwin - специализированного генератора отчетов, который входит в поставку BPwin.

Опция Ordering (на отчете по стрелкам отсутствует) сортирует данные по какому-либо значению.

Опция Multi-Valued Format регулирует вывод полей в отчете при группировке данных:

- Repeating Group - детальные данные объединяются в одно поле, между значениями вставляется +.
- Filled - дублирование данных для каждого заголовка группы;
- Header (опция по умолчанию) - печатается заголовок группы, затем - детальная информация.