

Лабораторная работа № 2

Тема. Создание диаграмм IDEF0.

Цель. Создание в среде Ramus Educational функциональной модели системы в нотации IDEF0.

Ход работы

1. Ознакомиться с теоретической частью.
2. Выполнить практическое задание.
3. Ответить на контрольные вопросы.
4. Оформить отчет.

Теоретическая часть

1. Общие сведения

Технология создания информационных систем (далее – ИС) предъявляет особые требования к методикам реализации и программным инструментальным средствам.

Реализацию проектов по созданию ИС принято разбивать на стадии анализа (прежде чем создавать ИС, необходимо понять и описать бизнес-логику предметной области), проектирования (необходимо определить модули и архитектуру будущей системы), непосредственного кодирования, тестирования и сопровождения. Известно, что исправление ошибок, допущенных на предыдущей стадии, обходится примерно в 10 раз дороже, чем на текущей, откуда следует, что наиболее критическими являются первые стадии проекта. Поэтому крайне важно иметь эффективные средства автоматизации ранних этапов реализации проекта.

Проект по созданию сложной ИС невозможно реализовать в одиночку. Коллективная работа существенно отличается от индивидуальной, поэтому при реализации крупных проектов необходимо иметь средства координации и управления коллективом разработчиков.

Вследствие значительного жизненного цикла может оказаться, что в процессе создания системы внешние условия изменились. Обычно внесение изменений в проект на поздних этапах создания ИС – весьма трудоемкий и дорогостоящий процесс. Поэтому для успешной реализации крупного проекта необходимо, чтобы инструментальные средства, на которых он реализуется, были достаточно гибкими к изменяющимся требованиям. На современном рынке средств разработки ИС достаточно много систем, в той или иной степени удовлетворяющих перечисленным требованиям.

CASE-средство верхнего уровня Ramus, поддерживающее методологии IDEF0 (функциональная модель) и DFD (Data Flow Diagram), предназначено для проведения анализа и реорганизации бизнес-процессов. Функциональная модель предназначена для описания существующих бизнес-процессов на предприятии (так называемая модель AS-IS) и идеального положения вещей – того, к чему нужно стремиться (модель TO-BE). Методология IDEF0 предписывает построение иерархической системы диаграмм – единичных описаний фрагментов системы. Сначала проводится описание системы в целом и ее взаимодействия с окружающим миром (контекстная диаграмма), после чего проводится функциональная декомпозиция – система разбивается на подсистемы и каждая подсистема описывается отдельно (диаграммы декомпозиции). После каждого сеанса декомпозиции проводится сеанс экспертизы: каждая диаграмма проверяется экспертами предметной области. Такая технология создания модели позволяет построить модель, адекватную предметной области на всех уровнях абстрагирования. Если в процессе моделирования нужно осветить специфические стороны технологии предприятия, Ramus позволяет переключиться на любой ветви модели на нотацию DFD и создать смешанную модель. Нотация DFD включает такие понятия, как "внешняя сущность" и "хранилище данных", что делает ее более удобной (по сравнению с IDEF0) для моделирования документооборота.

На основе модели Ramus можно построить модель данных. Для построения модели данных можно использовать такие инструменты как DBDesigner Fork или WWW SQL Designer. DBDesigner Fork поддерживает несколько уровней представления модели, в том числе, логический, что позволяет наглядно представить модель данных даже для неспециалистов. DBDesigner Fork позволяет проводить процессы прямого и обратного проектирования баз данных/

2. Создание моделей бизнес-процессов с использованием программного средства RAMUS

При запуске «Ramus» появляется окно, в котором

предлагается создать новый проект (по умолчанию) или же открыть уже существующий файл проекта. Данное окно не будет выводиться в дальнейшем если поставить галочку «Использовать выбор по умолчанию и больше не спрашивать».

При выборе пункта «Создать новый файл» автоматически запускается мастер создания нового проекта. Этот мастер можно закрыть и заполнить необходимые сведения позже, но рекомендуется выполнить все его этапы.

На первом этапе предлагается внести сведения об авторе, названии проекта и модели. Также следует выбрать тип нотации модели: IDEF0 или DFD. На втором этапе предлагается внести сведения о том, в какой организации используется данный проект. На третьем этапе предлагается дать короткое описание проекту. На четвертом этапе предлагается создать несколько основных классификаторов проекта. Например: «Информация», «Ресурсы» и т.д. Так как модели процессов реальных предприятий могут содержать значительное количество объектов (документы, персонал, функции и т.д.), то в Ramus предусмотрена возможность упорядочено хранить информацию об этих объектах в виде системы классификаторов. Классификация объектов упрощает поиск и обработку информации об объектах модели, а так же и об объектах непосредственно не представленных на диаграммах процессов, но относящихся к процессам предприятия. На пятом, заключительном этапе, предлагается выбрать какие классификаторы, из созданных, будут содержать перечень собственников процессов.

После завершения работы мастера, откроется рабочее пространство «Диаграммы» в котором можно приступить к созданию графической модели.

1.2 Создание модели в стандарте IDEF0

На начальных этапах создания ИС необходимо понять, как работает организация, которую собираются автоматизировать. 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.

В IDEF0 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 IDEF0 понимают описание системы (текстовое и графическое), которое должно дать ответ на некоторые заранее определенные вопросы. Процесс моделирования какой-либо системы в IDEF0 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.

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. 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.

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 трудоемким процессом (так называемая проблема "плавающей области").

Цель моделирования. Модель не может быть построена без

четко сформулированной цели. Цель должна отвечать на следующие вопросы:

- Почему этот процесс должен быть смоделирован?
- 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 ИС" и т. д.

Точка зрения. 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.

IDEF0-модель предполагает наличие четко сформулированной цели, единственного субъекта моделирования и одной точки зрения.

Модели AS-IS и TO-BE. Целью построения функциональных моделей обычно является выявление наиболее слабых и уязвимых мест деятельности организации, анализ преимуществ новых бизнес-процессов и степени изменения существующей структуры организации бизнеса. Анализ недостатков и "узких мест" начинают с построения модели AS-IS (Как есть), т. е. модели существующей организации работы. Исправление недостатков, перенаправление информационных и материальных потоков приводит к созданию модели TO-BE (Как будет) – модели идеальной организации бизнес-процессов. Как правило, строится несколько моделей TO-BE, среди которых определяют наилучший вариант.

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. 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.

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, создаваемая в Ramus может содержать два типа диаграмм:

- контекстную (в каждой модели может быть только одна контекстная диаграмма);
- декомпозиции.

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. Синтаксис описания системы в целом и каждого ее фрагмента одинаков во всей модели.

Основными компонентами нотации IDEF0, используемые при построении моделей, являются функциональные блоки и стрелки.

Функциональные блоки обозначают поименованные процессы, функции или задачи, которые происходят в течение определенного времени и имеют распознаваемые результаты. Функциональные блоки изображаются в виде прямоугольников. Все функциональные блоки должны быть названы и определены. Имя функционального блока должно быть выражено сочетанием отлагольного существительного, обозначающего процесс, например: «Работа с клиентом», «Прием заказа» и т. д. Функциональный блок «Работа с клиентом» может иметь, например, следующее определение: «Работа относится к полному циклу обслуживания клиента от приема звонка до оформления документов по заказу». При создании новой модели автоматически создается контекстная диаграмма на которой можно разместить единственный функциональный блок, изображающий систему в целом (рис. 1), щелкнув сначала по кнопке  на палитре инструментов, а затем по свободному месту на диаграмме. Для внесения имени функционального блока следует сделать по нему двойной щелчок, выбрать в редакторе свойств вкладку «Название» и внести имя блока.

Диаграммы декомпозиции содержат родственные работы, т. е. дочерние работы, имеющие общую родительскую работу. Для создания диаграммы декомпозиции следует щелкнуть по кнопке  (перейти к дочерним диаграммам). Возникает диалог «Создание новой диаграммы» (рис. 2), в котором следует указать нотацию новой диаграммы и функциональных блоков на ней. Появляется диаграмма декомпозиции (рис. 3).

Если оказывается, что количество функциональных блоков недостаточно, то функциональный блок можно добавить в диаграмму, щелкнув сначала по кнопке  на палитре инструментов, а затем по свободному месту на диаграмме.

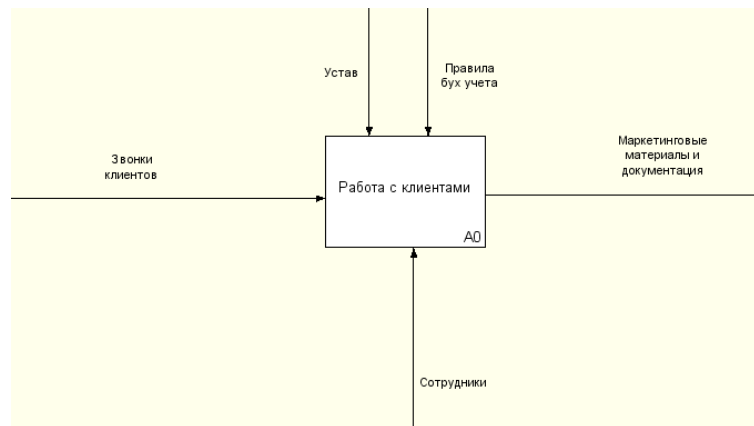


Рисунок 1 - Пример контекстной диаграммы

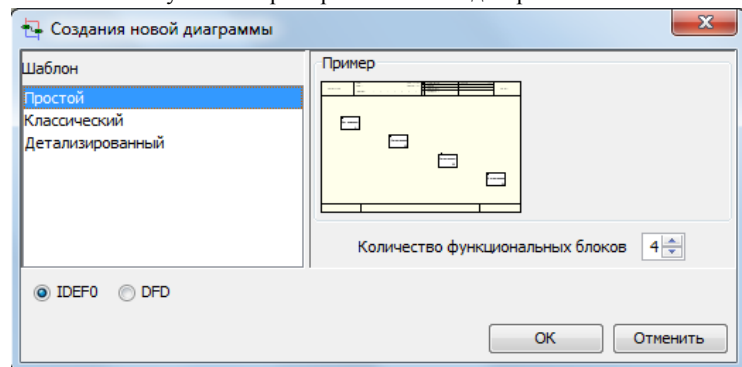


Рисунок 2 - Диалог «Создание новой диаграммы»

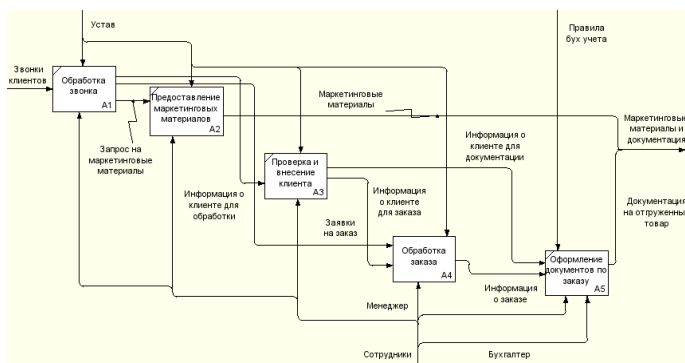


Рисунок 3 - Пример диаграммы декомпозиции

Функциональные блоки на диаграммах декомпозиции обычно располагаются по диагонали от левого верхнего угла к правому нижнему. Такой порядок называется порядком доминирования. Согласно этому принципу расположения в левом верхнем углу помещается самая важная работа или работа, выполняемая по времени первой. Далее вправо вниз располагаются менее важные или выполняемые позже работы. Такое расположение облегчает чтение диаграмм, кроме того, на нем основывается понятие взаимосвязей работ.

Взаимодействие работ с внешним миром и между собой описывается в виде **стрелок**. Каждая сторона функционального блока имеет стандартное значение с точки зрения связи блок/стрелка. В свою очередь, сторона блока, к которой присоединена стрелка, однозначно определяет ее роль.

Стрелки представляют собой некую информацию обозначаются существительными (например, «Устав», «Сотрудники», «Заказ») или именными сочетаниями (например, «Готовое изделие»).

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

Тоннелирование стрелок. Туннель - круглые скобки в начале и/или окончании стрелки. Туннельные стрелки означают, что данные, выраженные этими стрелками, не рассматриваются на родительской диаграмме и/или на дочерней диаграмме. Стрелка, помещенная в туннель там, где она присоединяется к блоку (рис. 4), означает, что данные, выраженные этой стрелкой, не обязательны на следующем уровне декомпозиции. Стрелка, помещаемая в туннель на свободном конце (рис. 5) означает, что выраженные ею данные отсутствуют на родительской диаграмме.

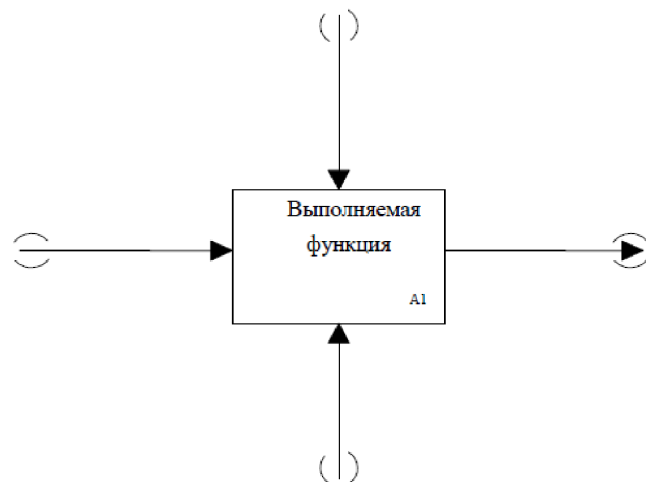
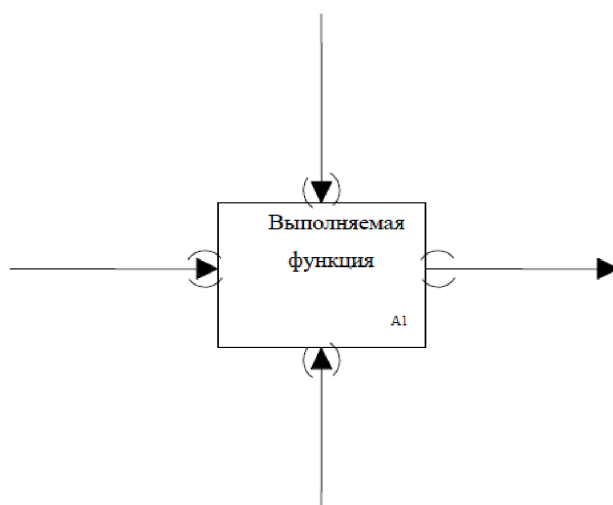


Рисунок 4 - Тоннелирование

Рисунок 5 - Тоннелирование «не в дочерней работе» «не в родительской диаграмме»

В программном комплексе Ramus вновь внесенные граничные стрелки на диаграмме декомпозиции нижнего уровня изображаются в квадратных скобках и автоматически не появляются на диаграмме верхнего уровня. Пункт «Туннель» позволяет использовать на диаграммах тоннелирование стрелок. Пункт доступен только при вызове контекстного меню стрелки на отрезке, обрaмлённом скобками. При выборе данного пункта откроется небольшое диалоговое окно с возможными вариантами действий.

В окне представлены два основных действия:

- «Создать стрелку» - позволяет отменить тоннелирование и создать отрезок стрелки на родительской диаграмме или внутри функционального блока (зависит от расположения туннеля);
- «Обозначить стрелку круглыми скобками» - позволяет, собственно, тоннелировать стрелку.

В случае если тоннелируются стрелки на входе в функциональный блок (стрелки управления, входов, механизмы) то доступен и третий вариант: «Обозначить стрелку круглыми скобками (показывать в отчётах для дочерних элементов)». Выбор данного варианта обозначает, что данная стрелка будет считаться стрелкой управления, входа или механизма соответственно, для всех дочерних функциональных блоков данного функционального блока, хотя на диаграммах этого и не будет показано.

Данную функциональность следует использовать только в крайних случаях, с целью повышения читабельности диаграмм, если на диаграммах присутствует множество объектов затрудняющих их восприятие.

Граничные стрелки. Следует также отметить, что программное обеспечение Ramus позволяет создавать сквозные стрелки (потоки) вне основной иерархии.

Для этого нужно открыть две диаграммы, каждую в своей вкладке, которые не являются дочерними друг к другу. После того как будет открыто две диаграммы, можно кликнуть на начале или на конце стрелки, и выбрав вкладку с другой диаграммой, присоединить стрелку к нужному функциональному блоку. В результате стрелка будет автоматически затоннелирована должным образом на обеих диаграммах, а возле значка туннеля на каждой диаграмме будет указан код другой диаграммы, на которой представлена начинается/заканчивается стрелка. Для обозначения такого типа туннелей используется специальный значок в виде круга, а не в виде скобок, как у обычного туннеля.

Несвязанные граничные стрелки. 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 Ramus как синтаксическая ошибка.

Для связывания стрелок входа, управления или механизма необходимо перейти в режим редактирования курсора, щелкнуть по

наконечнику стрелки и щелкнуть по соответствующему сегменту работы. Для связывания стрелки выхода необходимо перейти в режим курсора, 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 IDEF0 различают пять типов связей работ.

Связь по входу, когда стрелка выхода вышестоящей работы (далее - просто выход) направляется на вход нижестоящей

Связь по управлению, когда выход вышестоящей работы направляется на управление нижестоящей. Связь по входу показывает доминирование вышестоящей работы. Данные или объекты выхода вышестоящей работы не меняются в нижестоящей. На рис. 6 стрелка «Учебный план» связывает работы «Разработка учебного плана» и «Проведение занятий», при этом учебный план не претерпевает изменений в процессе проведения занятий.

Обратная связь по входу, когда выход нижестоящей работы направляется на вход вышестоящей. Такая связь, как правило, используется для описания циклов. На рис. 7 стрелка «Заказ на доукомплектование» связывает работы «Комплектование заказа» и «Проверка заказа», при этом выявленный недоукомплектованный заказ направляется на повторное комплектование.

Обратная связь по управлению, когда выход нижестоящей работы направляется на управление вышестоящей (стрелка «Рекомендации», рис. 8). Обратная связь по управлению часто свидетельствует об эффективности бизнес-процесса.



Рисунок 6- Связь по управлению

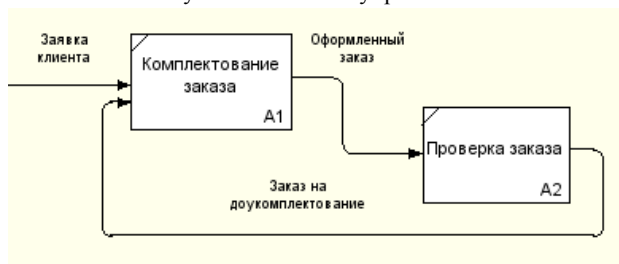


Рисунок 7 - Обратная связь по входу

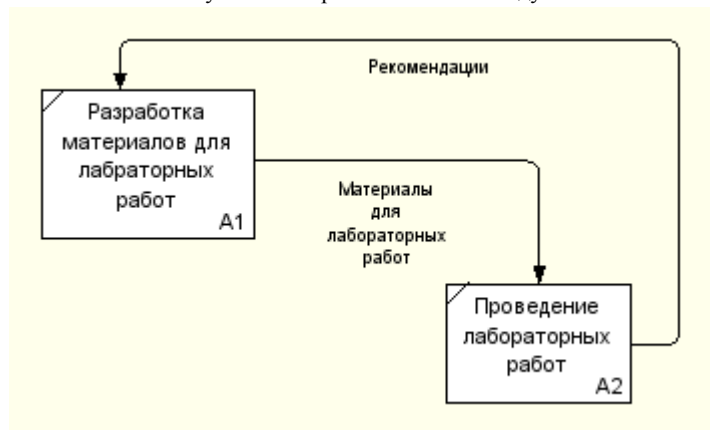


Рисунок 8 - Обратная связь по управлению

Связь выход-механизм, 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.

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. Правила именования таких стрелок рассмотрим 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. 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. 9). Правила именования сливающихся стрелок полностью аналогичны.

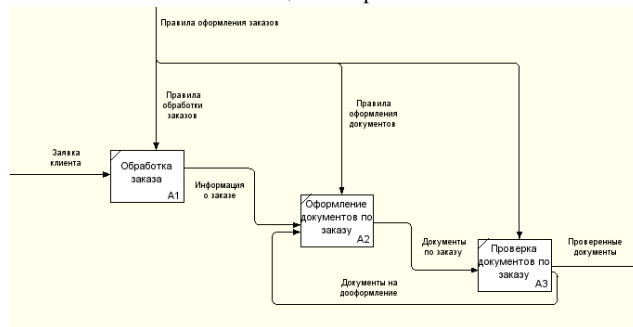


Рисунок 9 - Пример именования разветвляющейся стрелки

1.3 Нумерация работ и диаграмм

Все работы модели нумеруются. Номер состоит из префикса и числа. Может быть использован префикс любой длины, но обычно используют 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 порядковый 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. Такое дерево называют деревом узлов, а вышеописанную нумерацию – нумерацией по узлам.

Диаграммы IDEF0 имеют двойную нумерацию. Во-первых, диаграммы имеют номера по узлу. Контекстная диаграмма всегда имеет номер А-0, декомпозиция контекстной диаграммы – номер А0, остальные диаграммы декомпозиции – номера по соответствующему узлу (например, А1, А2, А21, А213 и т.д.). Ramus автоматически поддерживает нумерацию по узлам, т. е. 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. Ramus позволяет иметь в модели только одну диаграмму декомпозиции в данном узле. Прежние версии диаграммы можно хранить в виде бумажной копии.

1.4. Проведение экспертизы

Цикл автор-читатель. Этот цикл 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 профилей – 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 «Замечания», зачеркнуть цифру, соответствующую номеру замечания (это делается на твердой копии диаграммы).

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 диаграммы. Когда автор считает, что диаграмма уже достаточно проработана и достигла уровня «Рекомендовано», 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 «Публикация» и может быть распечатана и распространена среди участников проекта.

1.5. Пример выполнения практического задания

Описание предметной области «Формирование заказов»

Организация, заказчик разработки, занимается продажей различных товаров по заказам.

Деятельность фирмы организована следующим образом: склад получает товар под конкретный заказ, т.е. при приеме заказа от клиента определяется вид необходимой продукции и срок доставки на склад.

Такой способ приема заказов характерен для небольших фирм, которые хотят избежать затоваривания склада и продавать наиболее современные товары.

В силу данного обстоятельства требуется не только формирование заказа контракта и счета клиента, но и формирование заявки для доставки соответствующих товаров на склад

На основании приведенного описания предметной области, используя методологию IDEF0, можно построить следующую функциональную модель, описывающую основные

бизнес-процессы (рис. 10 - 12).

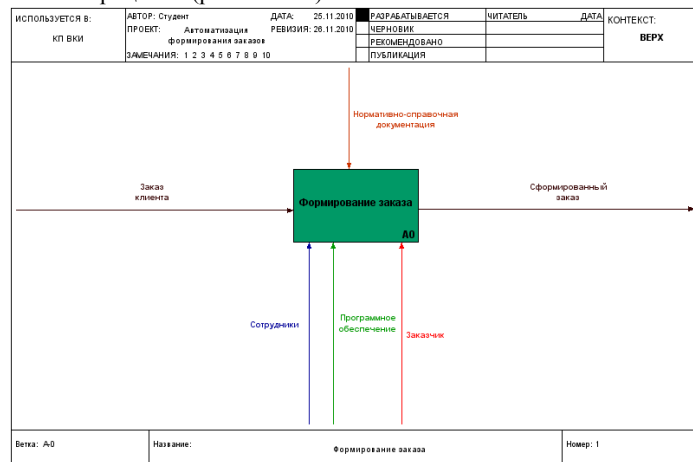


Рисунок 10 - Контекстная диаграмма

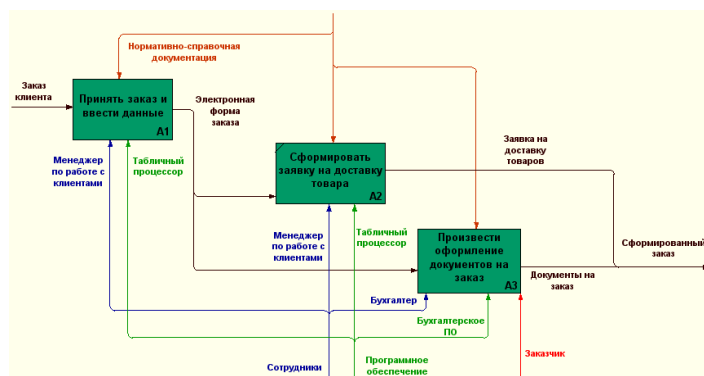


Рисунок 11 - Диаграмма декомпозиции 1-го уровня

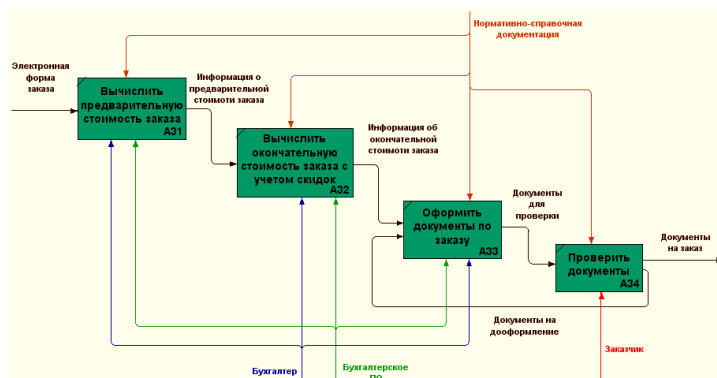


Рисунок 12 - Диаграмма декомпозиции 2-го уровня (процесс «Произвести оформление документов на заказ»)

Практическая часть

- Задание 1. Описать предметную область согласно варианту. Составить IDEF0 диаграмму. Отредактируйте внешний вид диаграммы (стрелки должны иметь разный цвет).
- Задание 2. Выполнить декомпозицию диаграмм до третьего уровня.
- Задание 3. Составить Ег-диаграмму

Контрольные вопросы

1. В чем заключается методология SADT?
2. Опишите нотацию IDEF0.
3. Поясните суть явления «плавающей области».
4. Какие типы связей между работами существуют?
5. Какие CASE-средства поддерживают методологию SADT и нотацию IDEF0?

Варианты заданий

1. Avto.ru (подбор автомобиля)
2. <https://www.avito.ru/>
3. <https://uchi.ru>
4. <http://www.consultant.ru/>
5. <https://uchi.ru>
6. <https://www.kinopoisk.ru/>

7. <https://stepik.org/>
8. <https://e.lanbook.com/>
9. <https://www.booking.com/index.ru.html>
10. <https://quizizz.com/>
11. <https://www.nava-td.ru/>
12. <https://www.222-2-222.ru/>
13. Свой вариант.