

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Дмитриев Николай Николаевич

Должность: Ректор

Дата подписания: 24.02.2025 04:07:45

Уникальный программный ключ:

f7c6227919e4cdbfb4d7b682991f8553b37cafbd

Министерство сельского хозяйства Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
Иркутский Государственный Аграрный Университет им. А.А.
Ежевского

Кафедра информатики и математического моделирования

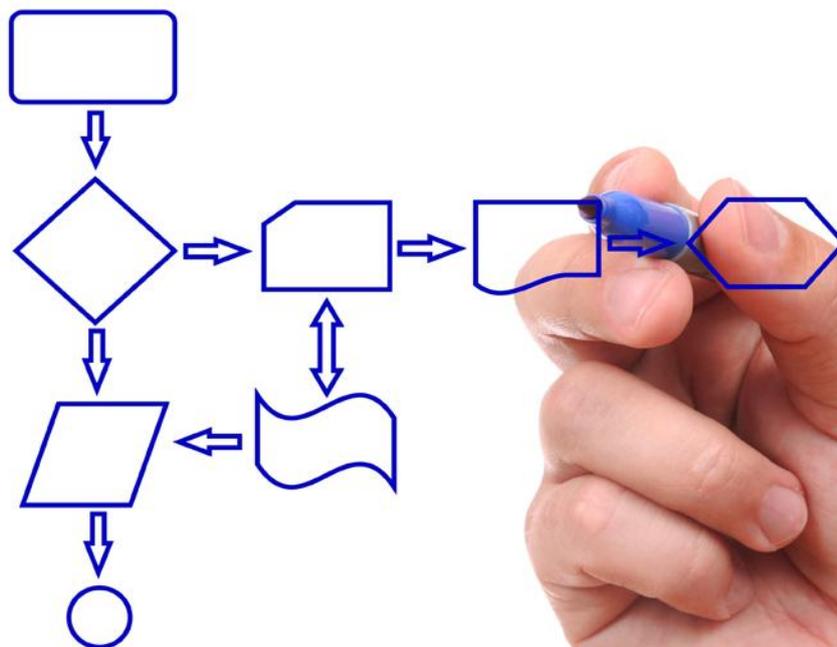
АСАЛХАНОВ П.Г., БЕНДИК Н.В.

Учебно-методическое пособие
«Проектирование информационных систем. Структурный подход»

для студентов

направления подготовки 09.03.03 «Прикладная информатика», профиль

«Прикладная информатика в экономике»



Иркутск, 2018

Печатается по решению научно-методического совета ФГБОУ ВО Иркутского ГАУ (протокол № от _____ 2018 г.).

Рецензенты:

Асалханов, П.Г. Проектирование информационных систем. Структурный подход / Учебно-методическое пособие для студентов направления «Прикладная информатика» // П.Г. Асалханов, Н.В. Бендик, – Иркутск: Изд-во Иркутский ГАУ, 2018. – 133 с. – ил.

Учебно-методическое пособие содержит материал по курсу "Проектирование информационных систем" и представляет собой практическое руководство по созданию информационных систем с использованием структурного подхода с помощью популярных CASE-средств. Пособие содержит описание методов структурного анализа и проектирования информационных моделей в объеме необходимом для практической работы. Подробно на конкретных примерах рассмотрено применения CASE-технологий и CASE-средств для автоматизации этапов анализа, проектирования и генерации кода информационных систем. Пособие предназначено для студентов, изучающих основы системного анализа и проектирования информационных систем.

© Асалханов П.Г., 2018

© Бендик Н.В., 2018

© Иркутский ГАУ, 2018

Содержание

Введение.....	4
1 Основные понятия технологии проектирования информационных систем..	6
1.1 Основные понятия и определения	6
1.2 Исторические аспекты развития технологий проектирования информационных систем	7
1.3 Процессы и модели жизненного цикла информационных систем.....	12
1.4 Основные методологии современного проектирования информационных систем	24
2 Организация проектирования информационных систем.....	28
2.1 Каноническое проектирование информационных систем	28
2.2 Стадии и этапы процесса канонического проектирования ИС.....	29
2.3 Типовое проектирование ИС, типовое проектное решение (ТПР)	35
3 Практикум по проектированию информационных систем.....	40
Лабораторная работа № 1. Основы работы в VPwin 4.1	40
Лабораторная работа № 2. Декомпозиция контекстной диаграммы.....	63
Лабораторная работа № 3. Построение функциональной модели. Описание модели	74
Лабораторная работа № 4. Построение диаграммы дерева узлов и FEO	84
Лабораторная работа № 5. Построение диаграммы потоков данных	94
Лабораторная работа № 6. Создание диаграммы IDEF3	103
Лабораторная работа № 7. Стоимостный анализ	118
Литература	131

Введение

Информация в современном мире превратилась в один из наиболее важных ресурсов, а информационные системы (ИС) стали необходимым инструментом практически во всех сферах деятельности.

Анализ современного состояния рынка ИС показывает устойчивую тенденцию роста спроса на информационные системы организационного управления. Причем спрос продолжает расти именно на интегрированные системы управления [10].

Тенденции развития современных информационных технологий приводят к постоянному возрастанию сложности ИС, создаваемых в различных областях деятельности.

Для успешной реализации ИС, объект проектирования должен быть адекватно описан, должны быть построены полные и непротиворечивые функциональные и информационные модели ИС. Накопленный к настоящему времени опыт проектирования ИС показывает, что это логически сложная, трудоемкая и длительная по времени работа, требующая высокой квалификации участвующих в ней специалистов [7]. Все это способствует появлению программно-технологических средств специального класса – CASE-средств (Computer Aided Software Engineering), реализующих CASE-технология создания и сопровождения ИС. CASE-технология представляет собой методологию проектирования ИС, а также набор инструментальных средств, позволяющих в наглядной форме моделировать предметную область, анализировать эту модель на всех этапах разработки и сопровождения ИС и разрабатывать приложения в соответствии с информационными потребностями пользователей. Большинство существующих CASE-средств основано на методологиях структурного или объектно-ориентированного анализа и проектирования, использующих спецификации в виде диаграмм или текстов для описания внешних

требований, связей между моделями системы, динамики поведения системы и архитектуры программных средств [7].

Учебно-методическое пособие является компиляцией популярных материалов, раскрывающих практические аспекты применения CASE-средств визуального проектирования с использованием структурного подхода, таких профессиональных авторов и практиков в этой области как: Маклаков С.В., Леоненков А.В., Вендров А.М., Грекул В.И., Денищенко Г.Н., Коровкина Н.Л.

В первой главе приводятся теоретические положения, связанные с нормативно-технической документацией на разработку и проектирование ИС, управление жизненным циклом ИС, внедрением и сопровождением ИС, а также получения практических навыков разработки основных проектных документов, моделирования и анализа бизнес-процессов, применения современных CASE-средств.

В второй главе учебно-методического пособия рассматривается ведущий инструмент визуального моделирования бизнес-процессов AIFusion Process Modeler (BPwin). Дано описание методологии и инструментальных средств, а также набор практических заданий, позволяющих освоить технику создания функциональных моделей. Последовательно рассматриваются три нотации моделирования, поддерживаемые BPwin: IDEF0, IDEF3 и DFD.

1 Основные понятия технологии проектирования информационных систем

1.1 Основные понятия и определения

Основные понятия в последние годы не претерпели сильных изменений, формулировки стали более точными и лаконичными, исключая неоднозначность понятий. Наиболее полные определения представлены в Федеральных законах Российской Федерации и стандартах.

Информация – «сведения (сообщения, данные) независимо от формы их представления» [7].

Информационные технологии – «процессы, методы поиска, сбора, хранения, обработки, предоставления, распространения информации и способы осуществления таких процессов и методов» [7].

Информационная система – «совокупность содержащейся в базах данных информации и обеспечивающих ее обработку информационных технологий и технических средств» [7].

Проектирование информационных систем – это упорядоченная совокупность методологий и средств создания или модернизации информационных систем.

Управление информационными системами – «применение методов управления процессами планирования, анализа, дизайна, создания, внедрения и эксплуатации информационной системы организации для достижения ее целей» [4].

Жизненный цикл информационных системы – «развитие рассматриваемой системы во времени, начиная от замысла и кончая списанием» [8].

Модель жизненного цикла – «структурная основа процессов и действий, относящиеся к жизненному циклу, которая служит в качестве общей ссылки для установления связей и взаимопонимания сторон» [8].

Архитектура информационных систем – это концепция, определяющая модель, структуру, выполняемые функции и взаимосвязь компонентов информационной системы.

Бизнес-процесс – это цепочка взаимосвязанных действий, направленных на создание товарной продукции или услуги.

Регламент бизнес-процесса – это четко определенный порядок выполнения бизнес-процесса, определяющий состав и действия участников.

Модель данных – это система организации данных и управления ими.

Методология проектирования информационных систем – это совокупность принципов проектирования (моделирования), выраженная в определенной концепции.

Средства моделирования – это программы описания и моделирования систем.

Типовое проектное решение (ТПР) – это многократно используемое проектное решение.

Нотации – это определенные способы представления элементов информационной системы.

Реинжиниринг бизнес-процессов – это фундаментальная реорганизация бизнес-процессов с целью повышения их эффективности.

Системный подход – процесс рассмотрения любой системы в качестве совокупности взаимосвязанных элементов.

Процессный подход – представление любой системы в качестве совокупности процессов.

Функциональный подход – предусматривает четкое закрепление за каждой структурной единицей набора функций.

Техническое задание – документ, используемый заказчиком в качестве средства для описания и определения задач, выполняемых при реализации договора [9].

1.2 Исторические аспекты развития технологий проектирования информационных систем

Середина прошлого столетия ознаменовалась началом активного развития информационных технологий. Прежде всего, военные ведомства и передовые предприятия многих стран понимают важность и ценность

создания и развития информационных систем. С появлением вычислительной техники обработка больших объемов информации и автоматизация основных производственных процессов и органов управления на всех уровнях становятся наиважнейшей задачей для обеспечения военного превосходства наиболее развитых государств и конкурентных преимуществ коммерческих компаний. Разработчики национальных и крупномасштабных информационных систем стали первыми осознавать необходимость создания специальных средств проектирования и моделирования бизнес-процессов, позволяющими сделать их работу более эффективной и сократить не только сроки создания информационных систем, но и минимизировать ошибки. Ошибки и неточности встречаются постоянно, чем раньше они диагностируются и локализуются, тем меньше стоимость переделки. Известно, что стоимость выявления и устранения ошибки на стадии проектирования в два раза обходится дороже, на стадии тестирования информационной системы в десять раз, а на стадии эксплуатации в сто раз, чем на стадии анализа бизнес-процессов и разработки технического задания.

При создании сложных информационных систем зачастую очень трудно понять требования персонала заказчика. Они могут быть сформулированы некорректно, а в процессе анализа конкретных бизнес-процессов даже измениться. Поэтому появление методологий современного проектирования и моделирования информационных систем было насущной задачей, над которой работали специалисты разных стран.

Появление автоматизированных систем управления в шестидесятых годах прошлого столетия определялось получением начальных знаний и опыта их разработки и внедрения. Анализировались все успехи и неудачи создания первых АСУ, но бесспорным было сокращение времени обработки информации, производственных и управленческих затрат и как следствие персонала.

Опыт зарубежных компаний по разработке и внедрению корпоративных информационных систем свидетельствует о появлении программ, в первую очередь связанных с автоматизацией учетных функций бухгалтерий, отдела кадров и складов. И намного позже появляются другие

автоматизированные системы управления производством, логистикой, взаимоотношениями с клиентами и поставщиками. На последнем этапе разрабатываются информационные системы управления всей компанией, позволяющие полностью перейти к электронному документообороту и автоматизировать все сферы деятельности фирмы. С появлением персональных компьютеров происходит децентрализация процессов управления, все чаще внедряются модули с распределенными системами обработки информации.

На следующем этапе, в семидесятых годах прошлого столетия пришло понимание, что информация – стратегический ресурс любой компании, который необходимо грамотно использовать. В тоже время главными потребителями информации являются руководители. Идеи использования распределенных систем не находят пока применения из-за отсутствия компактной вычислительной техники, которая появится позднее и перевернет весь мир. В компаниях и организациях создаются информационные отделы и службы, вычислительные центры и лаборатории.

Восьмидесятые годы характеризуются появлением специализированных методологий проектирования информационных систем и CASE-средств. На их основе разрабатываются первые программные средства, а персональные компьютеры позволяют приступить к созданию децентрализованных информационных систем. Различные персональные компьютеры объединяются в локальную сеть. Этот период характеризуется интеграцией информационных систем и появлением различных концепций управления ими на единой методологической основе.

Девяностые годы стали триумфом персональных компьютеров. Невысокая стоимость и компактные размеры сделали их чрезвычайно популярными и общедоступными для индивидуализации использования при решении управленческих задач. Разрабатываются корпоративные информационные системы, реализующие принципы распределенной обработки данных. Становится возможным автоматизация всех отделов и служб компаний, а не только бухгалтерии. Появляются системы электронного документооборота, в том числе для предприятий с развитой

филиальной сетью в разных городах и регионах. Сокращаются сроки обработки данных, производственных, складских и прочих управленческих отчетов.

Появление и развитие методологий моделирования и проектирования информационных систем не было простым процессом. На всех этапах этого пути были талантливые, энергичные, необычайно трудолюбивые люди, которые вкладывали свои знания, силы, опыт, а порой и денежные средства для успешной реализации информационных проектов. Вот некоторые из них:

В СССР основателем и теоретиком автоматизированных систем управления был выдающийся ученый академик В.М. Глушков. Под его руководством в 1963-1964 годах в Институте кибернетики Академии наук СССР были начаты работы по созданию автоматизированных систем сбора, учета, обработки данных, оперативно-календарного планирования производства на базе отечественной вычислительной техники. При этом ставилась задача разработки универсальной АСУ, пригодной для внедрения на многих предприятиях страны. Опытная эксплуатация и апробация проходили на самых современных предприятиях государства, таких как Львовский телевизионный завод «Электрон» и Кунцевский радиозавод. Некоторые решения были признаны и за рубежом, так была предложена общая алгоритмическая схема последовательного анализа вариантов, включавшая в себя вычислительные методы динамического программирования (В. С. Михалевича и Н. З. Шора). В.В. Шкурба развил эту схему вместе с методами имитационного моделирования для решения задач упорядочения, в частности в теории расписаний и календарного планировании.

В 1970-1980-х годах информационные системы интегрировались в комплексные АСУ, решающие задачи автоматизированного проектирования новых изделий, технологической подготовки производства и автоматизации организационного управления предприятием. Комплексные АСУ были разработаны и внедрены на Ульяновском авиационном заводе и других предприятиях оборонного комплекса под руководством А. А. Морозова, В. И. Скурихина. Комплексные АСУ создавались научно-исследовательскими

институтами такими как, Всесоюзного объединения “Союзсистемпром” Минприбора СССР: ЦНИИТУ, г. Минск; ГНИПИ ВТ, г. Казань; НИИУМС, г. Пермь и др.

В США Дуглас Т. Росс (SoftTech, Inc) разработал язык АПТ для работы станков с ЧПУ, который в середине 60-х положил начало разработкам графических языков моделирования. А в 1969 году им же предложена методология SADT (IDEF0) для моделирования информационных систем средней сложности, в рамках программы ICAM (Интеграция компьютерных и промышленных технологий), проводимой департаментом Военно-Воздушных Сил США в рамках большого аэрокосмического проекта.

К концу двадцатого века было разработано несколько десятков методов моделирования сложных систем. Они все были разные по функциональным возможностям, но во многом имели схожие подходы к анализу и описанию предметной области. Возникла острая необходимость объединения удачных решений в одну методику, которая устраивала большую часть разработчиков информационных систем. В результате этих процессов был разработан язык UML.

У многих ведущих компаний, таких как Rational Software, Oracle Corporation, IBM, Microsoft, HewlettPackard, i-Logix, Texas Instruments и Unisys была четкая уверенность в необходимости создания подобного языка программирования. С этой целью был создан консорциум UML Partners, рабочую группу по семантике UML возглавил Крис Кобрин. Ведущую роль в создании унифицированного языка моделирования (UML) сыграли Гради Буч, Айвар Джекобсон и Джеймс Рамбо, работающие в компании Rational Software. Ими разработаны следующие методы объектного моделирования сложных информационных систем:

1. Метод объектного моделирования программного обеспечения сложных информационных систем (метод Буча).
2. Метод анализа требований к бизнес- приложениям (метод Джекобсона).
3. Метод анализа обработки данных в сложных информационных системах (метод Рамбо).

Предварительная версия 0.8 унифицированного метода программирования была выпущена в октябре 1995 года. Первая версия UML 0.9 вышла в июне 1996 году и получила мощную поддержку Группы OMG, занимающейся разработкой единых стандартов в сфере web-технологий, включающую в себя несколько сотен компаний, работающих в области IT-технологий и производстве компьютерной техники. Это позволило выпустить в первые года сразу несколько версий. Так, в 1997 году появляется сразу две версии UML 1.0 и UML 1.1. В 1998 году разработчиками представляется версия UML 1.2, в 1999 году версия UML 1.3, в 2001 году выходит версия UML 1.4, а в 2003 году версия UML 1.5. Эта версия и принимается в качестве международного стандарта ISO/IEC 19501-2005.

Сейчас наиболее популярна версия UML 2.4.1, вышедшая в 2011 году, которая тоже оформлена виде международных стандартов ISO/IEC 19505-1 и 19505-2. Для нее разработаны инструментальные средства поддержки и визуального программирования, осуществляющих прямую генерацию кода из моделей UML, в основном посредством языков программирования C++ и Java. Среди них программы Rational Rose и Visual Paradigm for UML.

В настоящее время методологии и средства моделирования бизнес-процессов, процессно-ориентированных методов анализа и проектирования информационных систем широко представлены как в России, так и в большинстве стран мира.

1.3 Процессы и модели жизненного цикла информационных систем

В соответствии с ГОСТ Р ИСО/МЭК 12207-99 процессы жизненного цикла включают себя работы, которые могут выполняться в жизненном цикле программных средств, распределены по пяти основным, восьми вспомогательным и четырем организационным процессам [15].

Основные процессы жизненного цикла

Основные процессы жизненного цикла состоят из пяти процессов, которые реализуются под управлением основных сторон, вовлеченных в жизненный цикл программных средств. Под основной стороной понимают

одну из тех организаций, которые инициируют или выполняют разработку, эксплуатацию или сопровождение программных продуктов. Основными сторонами являются заказчик, поставщик, разработчик, оператор и персонал сопровождения программных продуктов. Основными процессами являются:

1. **Процесс заказа.** Определяет работы заказчика, то есть организации, которая приобретает систему, программный продукт или программную услугу.

2. **Процесс поставки.** Определяет работы поставщика, то есть организации, которая поставляет систему, программный продукт или программную услугу заказчику.

3. **Процесс разработки.** Определяет работы разработчика, то есть организации, которая проектирует и разрабатывает программный продукт.

4. **Процесс эксплуатации.** Определяет работы оператора, то есть организации, которая обеспечивает эксплуатационное обслуживание вычислительной системы в заданных условиях в интересах пользователей.

5. **Процесс сопровождения.** Определяет работы персонала сопровождения, то есть организации, которая предоставляет услуги по сопровождению программного продукта, состоящие в контролируемом изменении программного продукта с целью сохранения его исходного состояния и функциональных возможностей. Данный процесс охватывает перенос и снятие с эксплуатации программного продукта.

Вспомогательные процессы жизненного цикла [9]

Вспомогательные процессы жизненного цикла состоят из восьми процессов. Вспомогательный процесс является целенаправленной составной частью другого процесса, обеспечивающей успешную реализацию и качество выполнения программного проекта. Вспомогательный процесс, при необходимости, инициируется и используется другим процессом. Вспомогательными процессами являются:

1. **Процесс документирования.** Определяет работы по описанию информации, выдаваемой в процессе жизненного цикла.

2. **Процесс управления конфигурацией.** Определяет работы по управлению конфигурацией.

3. **Процесс обеспечения качества.** Определяет работы по объективному обеспечению того, чтобы программные продукты и процессы соответствовали требованиям, установленным для них, и реализовывались в рамках утвержденных планов. Совместные анализы, аудиторские проверки, верификация и аттестация могут использоваться в качестве методов обеспечения качества.

4. **Процесс верификации.** Определяет работы (заказчика, поставщика или независимой стороны) по верификации программных продуктов по мере реализации программного проекта.

5. **Процесс аттестации.** Определяет работы (заказчика, поставщика или независимой стороны) по аттестации программных продуктов программного проекта.

6. **Процесс совместного анализа.** Определяет работы по оценке состояния и результатов какой-либо работы. Данный процесс может использоваться двумя любыми сторонами, когда одна из сторон (проверяющая) проверяет другую сторону (проверяемую) на совместном совещании.

7. **Процесс аудита.** Определяет работы по определению соответствия требованиям, планам и договору. Данный процесс может использоваться двумя сторонами, когда одна из сторон (проверяющая) контролирует программные продукты или работы другой стороны (проверяемой).

8. **Процесс решения проблемы.** Определяет процесс анализа и устранения проблем (включая несоответствия), независимо от их характера и источника, которые были обнаружены во время осуществления разработки, эксплуатации, сопровождения или других процессов.

Организационные процессы жизненного цикла [15]

Организационные процессы жизненного цикла состоят из четырех процессов. Они применяются в какой-либо организации для создания и реализации основной структуры, охватывающей взаимосвязанные процессы жизненного цикла и соответствующий персонал, а также для постоянного совершенствования данной структуры и процессов. Эти процессы, как правило, являются типовыми, независимо от области реализации конкретных

проектов и договоров; однако уроки, извлеченные из таких проектов и договоров, способствуют совершенствованию организационных вопросов. Организационными процессами являются:

1. **Процесс управления.** Определяет основные работы по управлению, включая управление проектом, при реализации процессов жизненного цикла.

2. **Процесс создания инфраструктуры.** Определяет основные работы по созданию основной структуры процесса жизненного цикла.

3. **Процесс усовершенствования.** Определяет основные работы, которые организация (заказчика, поставщика, разработчика, оператора, персонала сопровождения или администратора другого процесса) выполняет при создании, оценке, контроле и усовершенствовании выбранных процессов жизненного цикла.

4. **Процесс обучения.** Определяет работы по соответствующему обучению персонала.

Данный ГОСТ Р ИСО/МЭК 12207-99 охватывает жизненный цикл программных средств от концепции замыслов через определение и объединение процессов для заказа и поставки программных продуктов и услуг. Кроме того, данная структура предназначена для контроля и модернизации данных процессов.

Процессы, определенные в настоящем стандарте, образуют множество общего назначения. Конкретная организация, в зависимости от своих целей, может выбрать соответствующее подмножество процессов для выполнения своих конкретных задач. Поэтому настоящий стандарт следует адаптировать для конкретной организации, проекта или приложения. Настоящий стандарт предназначен для использования как в случае отдельно поставляемых программных средств, так и для программных средств, встраиваемых или интегрируемых в общую систему [15].

Модели жизненного цикла информационной системы

Модель жизненного цикла информационной системы может включать:

1. Стадии.
2. Основные результаты выполнения работ на каждой стадии.
3. Ключевые события.

Под стадией понимается определенный этап процесса разработки информационной системы.

Жизненный цикл информационной системы характеризуется периодом времени от идеи создания информационной системы и заканчивая моментом вывода ее из эксплуатации и включает в себя следующие стадии:

1. Предпроектное обследование.
2. Проектирование.
3. Создание информационной системы.
4. Ввод в эксплуатацию.
5. Эксплуатация информационной системы.
6. Вывод из эксплуатации.

Процессы жизненного цикла информационных систем представлены на рисунке 1.

Процессы жизненного цикла, определенные стандартом ГОСТ Р ИСО/МЭК 15288-2005, могут применяться любой организацией при приобретении и использовании или создании и поставке системы. Они распространяются на любой уровень системной иерархии и на любую стадию жизненного цикла.

Процессы жизненного цикла основываются на принципах модульности (максимальная слаженность функций процесса и минимальная связь между процессами) и собственности (процесс связывается с ответственностью). Функции, которые осуществляются данными процессами, определяются в зависимости от конкретных целей, результатов и набора действий, составляющих данный процесс. Процессы, описанные в настоящем стандарте, не препятствуют и не исключают использование дополнительных процессов, которые организация считает необходимыми [17].

Управление процессами жизненного цикла ИС

Цель управления процессами жизненного цикла системы заключается в гарантировании доступности эффективных процессов жизненного цикла для использования организацией. Данный процесс обеспечивает процессы жизненного цикла системы, которые согласованы с целями и политикой организации, определены, адаптированы и поддерживаются соответствующим

образом для учета особенностей отдельных проектов и способны реализовываться с помощью эффективных проверенных методов и инструментальных средств.

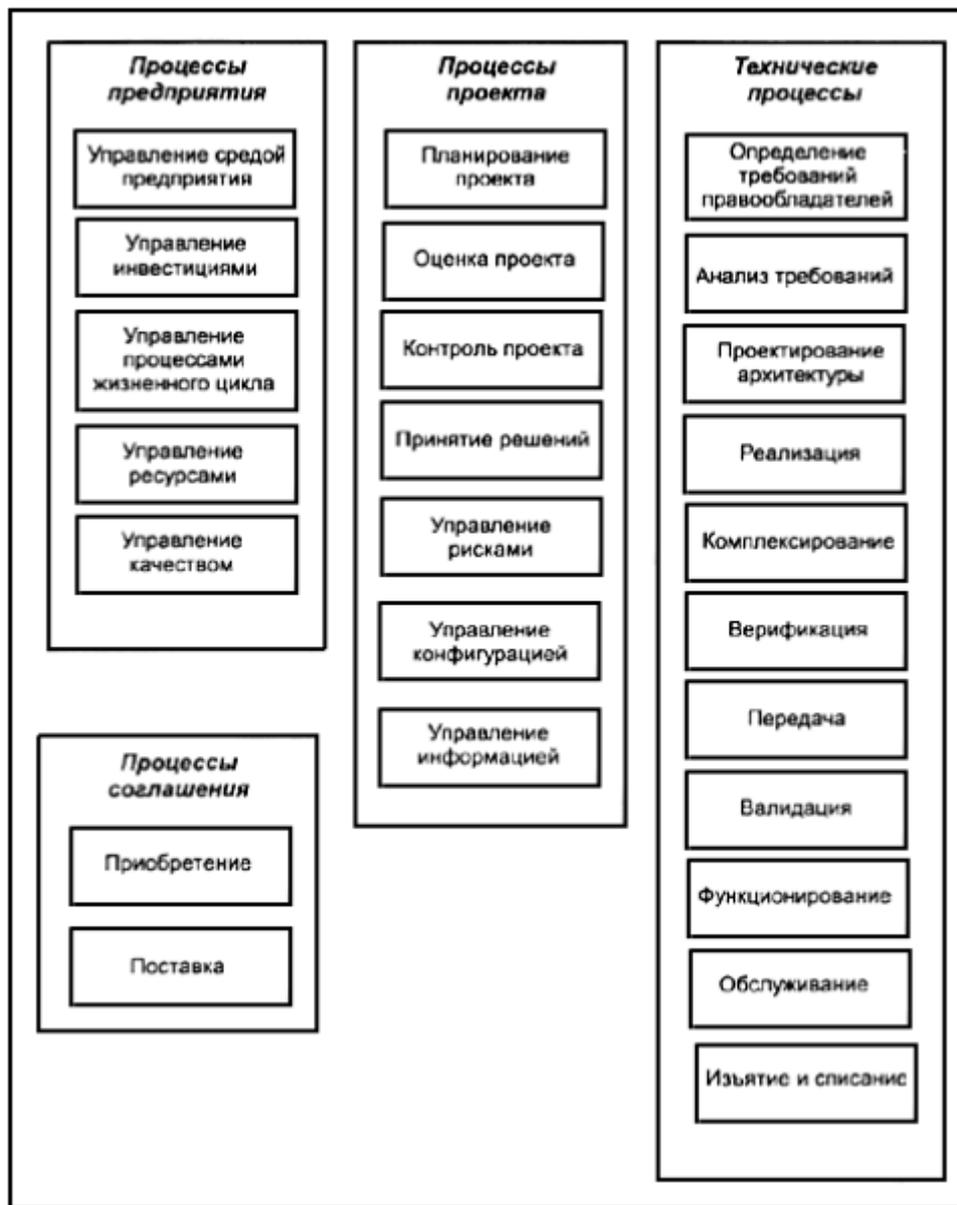


Рисунок 1 - Процессы жизненного цикла информационной системы

В результате эффективного управления процессами жизненного цикла системы:

- определяются процессы жизненного цикла системы, которые будут использоваться организацией;
- определяется политика применения процессов жизненного цикла системы;

с) определяется политика адаптации процессов жизненного цикла системы для удовлетворения потребностей отдельных проектов;

д) определяются критерии оценки результатов применения процессов жизненного цикла системы;

е) предпринимаются действия по совершенствованию способов определения и применения процессов жизненного цикла системы.

При реализации процессов управления процессами жизненного цикла системы организация должна осуществлять следующие действия в соответствии с принятой политикой и процедурами:

а) устанавливать стандартные наборы процессов жизненного цикла систем для соответствующих стадий жизненного цикла системы;

б) определять приемлемые политику и процедуры адаптации и требования к их утверждению;

с) определять методы и инструментальные средства, которые поддерживают выполнение процессов жизненного цикла системы;

д) по возможности устанавливать показатели, которые позволяют определять характеристики выполненных стандартных процессов;

е) контролировать выполнение процесса, сохранять и анализировать показатели процесса и определять тенденции по отношению к критериям предприятия;

ф) определять возможности для усовершенствования стандартных процессов жизненного цикла систем;

г) совершенствовать имеющиеся процессы, методы и инструментальные средства, используя найденные возможности.

Каждая система имеет свой жизненный цикл. Жизненный цикл может быть описан с использованием абстрактной функциональной модели, представляющей концептуализацию потребности в системе, ее реализации, применения, развития и ликвидации [17].

Система развивается на протяжении жизненного цикла в результате действий, осуществляемых и управляемых людьми, работающими в организациях и использующими определенные процессы в своей деятельности. Детали модели жизненного цикла выражаются в терминах этих

процессов, их результатов, взаимосвязи и возникновения. Настоящий стандарт определяет множество процессов, названных процессами жизненного цикла, при помощи которых может быть смоделирован жизненный цикл системы.

Жизненные циклы различаются по свойствам, целям, использованию системы, а также по преобладающим условиям. Тем не менее, несмотря на очевидное множество различий в жизненных циклах систем, существует базовый набор стадий жизненного цикла, составляющих полный жизненный цикл любой системы. Каждая стадия имеет определенную цель и вклад в полный жизненный цикл и рассматривается при планировании и выполнении жизненного цикла системы [17].

Стадии представляют собой основные периоды жизненного цикла, связанные с системой и относящиеся к состоянию описания системы или непосредственно к системе. Стадии отображают значимый прогресс и достижение запланированных этапов развития системы на протяжении всего жизненного цикла и дают начало важнейшим решениям относительно своих входов и выходов. Эти решения используются организациями для учета неопределенностей и рисков, непосредственно связанных с затратами, сроками и функциональностью при создании или применении системы. Таким образом, стадии обеспечивают организации структурой работ, в рамках которых управление предприятием обладает высокой способностью для обзора и контроля проекта и технических процессов.

Организации проходят стадии жизненного цикла различными способами, устраняя противоречия между стратегией осуществления бизнеса и стратегией уменьшения рисков. Параллельное прохождение стадий или их прохождение в различном порядке может привести к формам жизненного цикла с совершенно разными характеристиками.

Часто в качестве альтернативных вариантов используются последовательная, инкрементная или эволюционная формы жизненного цикла; в отдельных случаях могут быть разработаны комбинации этих форм. Выбор и разработка организацией конкретных форм жизненного цикла зависят от ряда факторов, включая бизнес-контекст, природу и сложность

системы, стабильность требований, технологические возможности, потребность в различных системных возможностях во времени и наличие бюджетных средств и ресурсов [17].

Аналогично тому, как все системные элементы осуществляют вклад в систему как в единое целое, так и каждая стадия жизненного цикла должна учитываться на любой другой ее стадии. Следовательно, участвующие стороны должны координировать свои действия и кооперироваться друг с другом на протяжении всего жизненного цикла. Синергия стадий жизненного цикла и сторон, вкладывающих средства в реализацию функциональностей на этих стадиях, является необходимой для успешного осуществления проектных мероприятий.

Тесная связь и, по возможности, единение проектных команд, различных функций и организаций, ответственных за другие стадии жизненного цикла, приводят к логичности и согласованности жизненного цикла [11]. Наибольшее распространение получили следующие модели жизненного цикла информационных систем: каскадная (классическая или водопадная), итерационная и спиральная.

Каскадная (классическая, водопадная) модель жизненного цикла информационной системы

Модель была предложена в 1970 году Уинстоном Ройсом. Переход на следующий этап осуществляется после полного окончания работ по предыдущему этапу, при этом оформляется полный комплект рабочей документации. Все этапы выполняются в строгой последовательности с утвержденными сроками и четкими затратами. Это основные достоинства каскадной модели ЖЦ ИС, которая применялась в условиях полной определенности решаемых задач и совершенно не приемлема, когда и разработчики, и заказчики не имеют четкого видения всех особенностей проектируемой ИС. Кроме того, невозможно идти дальше, пока не сдан предыдущий этап, а после сдачи нельзя возвращаться к нему для устранения обнаруженных недочетов, что серьезно затрудняет работы по совершенствованию и доработке, создаваемой ИС. Эта модель нравится и заказчикам, и разработчикам по причине жесткой дисциплины

финансирования этапов только после их предъявления. Но полностью отсутствует гибкость в работе над созданием ИС.

Каскадная модель представлена на рисунке 2. На практике, все же приходится возвращаться к предыдущим этапам и в этом случае, в последнее время наиболее востребованной стала итерационная модель ЖЦ ИС.



Рисунок 2 – Каскадная (водопадная, классическая) модель ЖЦ ИС

Итерационная модель жизненного цикла ИС

Поэтапная модель с промежуточным контролем — итерационная модель разработки информационной системы. Каждый этап имеет обратные связи в процессе корректировки и создает условия для корректировки ранее созданных этапов. При этом трудоемкость работ и временные затраты существенно сокращаются по сравнению с водопадной моделью жизненного цикла. Итерационная модель ЖЦ информационной системы представлена на рисунке 3.



Рисунок 3 – Итерационная модель ЖЦ ИС

Создание информационной системы – это организованный процесс построения и последовательного преобразования согласованных моделей на всех этапах жизненного цикла. При этом все разработанные модели находятся в репозитории проекта и доступны всем разработчикам, что позволяет эффективно вести одновременную работу над проектированием и созданием информационной системы.

Спиральная модель жизненного цикла информационной системы

Спиральная модель предложена Барри Боэм в 1988 году и определяет, в основном стартовые этапы жизненного цикла информационной системы. При этом обосновывается и проверяется возможность реализации спроектированных технических решений. На каждом витке создается прототип проектируемой информационной системы, который на следующих витках спирали ЖЦ ИС совершенствуется, дополняется и доводится до полного внедрения. При этом не обязательно дожидаться окончания каждого этапа, данная модель позволяет переходить на следующие витки спирали и решать проблемы или недоделки на следующем уровне, что делает работу над проектом более эффективной, гибкой и завершить в более сжатые сроки.

Новый виток спирали соответствует поэтапной модели создания фрагмента информационной системы. При использовании спиральной модели ЖЦ:

- происходит ориентация на модернизацию информационной системы;
- осуществляется аккумуляция всех решений в процессе проектирования и создания моделей и прототипов информационной системы;
- проводится анализ издержек и всех рисков в процессе проектирования ИС.

Спиральный процесс состоит из следующей повторяющейся последовательности:

1. Определение требований.
2. Анализ.
3. Проектирование.
4. Реализация и тестирование.
5. Интеграция.
6. Внедрение.

Этот многократный цикл, завершающийся созданием новой версии информационной системы представлен на рисунке 4.

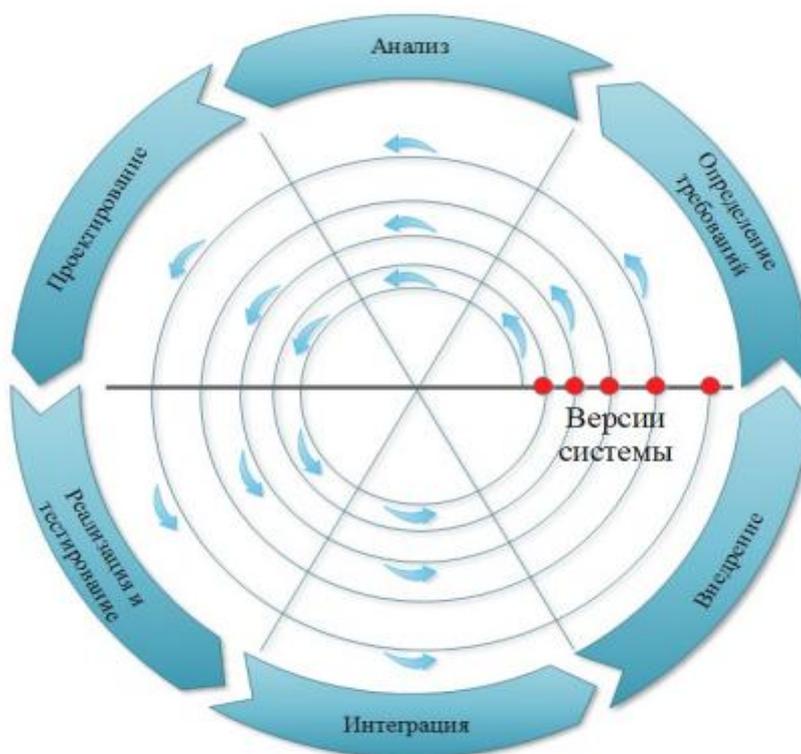


Рисунок 4 – Спиральная модель ЖЦ ИС

Для применения спиральной модели ЖЦ ИС может быть несколько причин, это необходимость минимизации рисков и возможность представления заказчику прототип или эскизную версию проекта для конкретизации пожеланий и учета их в следующих циклах. А также в случае если разрабатываемая информационная система достаточно сложна и существует реальная необходимость создавать промежуточные версии продукта, не откладывая эту работу на финишные этапы, как это предписывает водопадная модель.

Основная задача спиральной модели жизненного цикла информационной системы заключается в том, чтобы на каждой итерации создавать очередную версию системы, используя разработанный прототип предыдущих этапов. Такая модель позволяет более гибко работать с заказчиком, постоянно учитывать его замечания и предложения, совершенствовать проектируемую систему в процессе каждого нового витка спирали.

1.4 Основные методологии современного проектирования информационных систем

Методология функционального моделирования работ SADT

Методология SADT (Structured Analysis and Design Technique - методология структурного анализа и проектирования), разработанная Дугласом Т. Россом в 1969-1973 годах базируется на структурном анализе систем и графическом представлении организации в виде системы функций, которые имеют три класса структурных моделей:

1. Функциональная модель.
2. Информационная модель.
3. Динамическая модель.

Процесс моделирования по методологии SADT состоит из следующих этапов:

1. Сбор информации и анализ информации о предметной области.
2. Документирование полученной информации.

3. Моделирование (IDEF0).

4. Корректурa модели в процессе итеративного рецензирования.

Методология в настоящее время более известна как нотация IDEF0, использует формализованный процесс моделирования информационных систем и имеет следующие стадии: анализ, проектирование, реализация, объединение, тестирование, установка, функционирование. Проектирование информационных систем по стандарту IDEF0 сводится к декомпозиции основных функций организации на отдельные бизнес-процессы, работы или действия. В результате разрабатывается иерархическая модель анализируемой организации, при этом декомпозицию можно проводить многократно, до четкого и детального описания всех процессов. Диаграммы IDEF0 верхнего уровня принято называть родительскими, а нижнего уровня – дочерними.

Основным достоинством этой методологии являются простота и наглядность. В качестве недостатка – невозможность описать реакцию описываемого процесса на изменяющиеся внешние факторы. Для этих целей служат другие методологии.

Методология RAD – быстрой разработки приложений

Принципы RAD сформулированы в 1980 году сотрудником компании IBM Джеймсом Мартином. Они базировались на идеях Скотта Шульца и Барри Бойма при этом методология реализовывалась в кратчайшие сроки небольшой группой разработчиков с использованием инкрементного прототипирования. Это позволяло на ранней стадии проектирования ИС продемонстрировать заказчику действующую интерактивную модель системы-прототипа, уточнить проектные решения, оценить эксплуатационные характеристики.

В настоящее время методология RAD стала общепринятой схемой для проектирования и разработки информационных систем. Средства разработки, основанные на RAD, очень популярны за счет использования таких программных сред разработки: IBM Lotus Domino Designer, Borland Delphi, Borland C++ Builder, Microsoft Visual Studio, Macromedia Flash и др.

В методологии RAD быстрая разработка приложений достигается за счет использования компонентно-ориентированного конструирования и применяется если:

- ✓ Бюджет проектируемой информационной системы ограничен.
- ✓ Нечетко определены требования к информационной системе.
- ✓ Требуется реализация проекта информационной системы в минимальные сроки.
- ✓ Интерфейс пользователя можно продемонстрировать в прототипе.
- ✓ Проект можно разделить на составляющие элементы по функциональному назначению.

Методология RAD имеет следующие стадии:

1. Моделирование информационных потоков между бизнес-функциями.
2. Моделирование данных.
3. Преобразование объектов данных, обеспечивающих реализацию бизнес-функций.
4. Генерация приложений.
5. Тестирование и объединение.

Недостатки методологии RAD:

1. Для больших информационных систем требуются большой коллектив разработчиков.
2. Применима для информационных систем, которые могут декомпозироваться на отдельные модули и в которых производительность не является критической величиной.
3. Не используется в случае применения новых технологий.

Методология RUP

Среди всех фирм-производителей CASE-средств компания IBM Rational Software Corp. (до августа 2003 года – это самостоятельная фирма Rational Software Corp.) одна из первых осознала стратегическую перспективность развития объектно-ориентированных технологий анализа и проектирования программных систем. Эта компания выступила инициатором

унификации языка визуального моделирования в рамках консорциума OMG, что привело к появлению первых версий языка UML. Эта же компания первой разработала инструментальное объектно-ориентированное CASE-средство, в котором был реализован язык UML, как базовая нотация визуального моделирования. Графическое представление методологии RUP из Википедии изображено на рисунке 5.

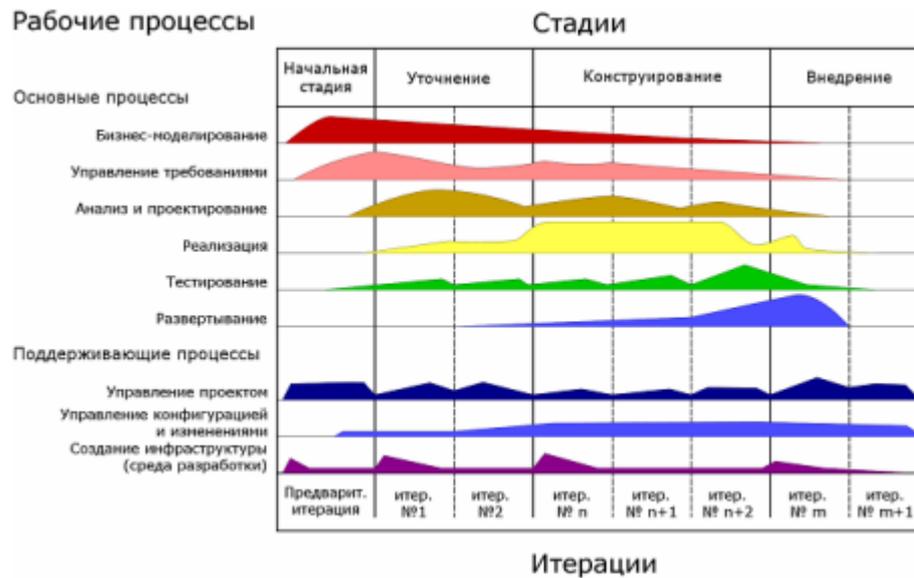


Рисунок 5 – Представление методологии RUP

Одна из самых популярных технологий - Rational Unified Process (RUP). В определенном плане эта методология становится международным стандартом, разработанный компанией Rational Software, которая в настоящее время входит в состав IBM. Авторами UML считаются сотрудники фирмы Rational Software: Гради Буч, Айвар Якобсон, Джемс Рамбо. RUP полностью соответствует стандартам, определяющим проектные работы в процессе жизненного цикла информационных систем. В методологии RUP реализуются следующие подходы:

1. Итерационный и инкрементный (наращиваемый).
2. Построение системы на базе архитектуры информационной системы.
3. Планирование и управление проектом на основе функциональных требований к информационной системе.

Разработка информационной системы выполняется итерациями. Это отдельные проекты небольшие по объему и содержанию, которые включают свои собственные этапы анализа требований, проектирования, реализации, тестирования, интеграции. Заканчиваются итерации созданием работающей информационной подсистемы.

Итерационный цикл характеризуется периодической обратной связью и может адаптироваться к ядру разрабатываемой системы. Создаваемая информационная система постепенно растет и совершенствуется.

2 Организация проектирования информационных систем

2.1 Каноническое проектирование информационных систем

Каноническое проектирование ИС направлено на отражение особенностей технологии индивидуального (оригинального) проектирования. Среди основных характерных особенностей канонического проектирования можно выделить такие особенности, как:

- отражение особенностей ручной технологии проектирования;
- ориентация на индивидуальное (оригинальное) проектирование;
- осуществление на уровне исполнителей; возможность интеграции выполнения элементарных операций; применение, как правило, для сравнительно небольших, локальных ИС;
- использование инструментальных средств универсальной компьютерной поддержки.

Каноническое проектирование направлено на минимальное использование типовых проектных решений. Адаптация проектных решений при каноническом проектировании осуществляется только путем перепрограммирования соответствующих программных модулей.

Организация канонического проектирования ИС основана на использовании каскадной модели жизненного цикла и предусматривает набор определенных стадий и этапов. Принцип деления процесса проектирования на стадии и этапы направлен на то, чтобы проектировать

систему «сверху-вниз» и постепенно разрабатывать - изначально укрупненные, затем детализированные – проектные решения.

Поскольку объекты автоматизации имеют различную сложность и набор задач для создания решения для конкретной ИС, стадии и этапы работ также могут различаться по трудоемкости: существует возможность объединять последовательные этапы, исключать определенные из них на любой стадии проекта, а также до окончания предыдущей стадии начинать выполнение следующей.

Стадии и этапы разработки ИС, которые выполняют организационные участники, оформляются в договорах и технических заданиях на выполнение работ. Каноническое проектирование основано на ряде стандартов, таких, как:

1. ГОСТ 34.003 – термины и определения основных понятий в области автоматизированных систем;
2. ГОСТ 34.201 – виды, комплектность и обозначение документов при создании автоматизированных систем;
3. ГОСТ 34.601 – стадии создания автоматизированных систем;
4. ГОСТ 34.602 – техническое задание на создание ИС;
5. ГОСТ 34.603 – виды испытаний автоматизированных систем;
6. РД 50-34.698 – требования к содержанию документов;
7. ГОСТ 2.105 – общие требования к текстовым документам.

По отношению к проекту разработки ИС можно выделить 3 укрупненные стадии проектирования:

- предпроектную (стадии 1-3);
- проектную (стадии 4-6);
- послепроектную (стадии 7-8).

2.2 Стадии и этапы процесса канонического проектирования ИС

Стадии и этапы создания ИС, выполняемые организациями-участниками, фиксируются в договорах и технических заданиях на выполнение работ.

Предпроектная стадия направлена на предпроектное обследование и разработку технического задания на ИС. Характерными результатами этого этапа являются: определение целей и задач системы, формирование общих требований к ее созданию, разработка программы проведения обследования, в ходе которого должны быть изучены структура и бизнес-процессы организации, модель управления, задачи, подлежащие автоматизации, технико-экономические характеристики, ориентировочных состав технических средств.

Перечень 8 этапов работ (стадий), в соответствии с ГОСТ 34.601 и дополнительными пояснениями, представлен ниже:

Стадия 1. Формирование требований к ИС.

- обследование объекта и обоснование необходимости создания ИС;
- формирование требований пользователя к ИС;
- оформление отчета о выполненной работе и заявки на разработку ИС (ТТХ).

Стадия 2. Разработка концепции ИС.

- изучение объекта;
- проведение необходимых научно-исследовательских работ;
- разработка вариантов концепции ИС, удовлетворяющих требованиям пользователей;
- оформление отчета о проделанной работе.

Стадия 3. Техническое задание.

- разработка и утверждение технического задания на создание ИС.

Важным документом, фиксирующим результаты определения стратегии внедрения ИС, является технико-экономическое обоснование проекта. В этом документе должно быть четко определены результаты выполнения проекта для заказчика, а также указаны графики выполнения работ и график финансирования на разных этапах выполнения проекта. Дополнительно в таком документе отражаются сроки, время окупаемости проекта, ожидаемые выгода и экономический эффект проекта.

Ориентировочно технико-экономическое обоснование содержит:

- все риски и ограничения, влияющие на успешность проекта;

- условия эксплуатации будущей системы: архитектурные, программные, аппаратные требования, требования к компонентам ПО и СУБД;

- пользователи системы;

- функции, выполняемые системой;

интерфейсы и распределение функций между человеком и системой;

сроки завершения этапов, форма приемки/ сдачи работ;

рамки проекта; возможности развития системы.

По результатам обследования формируется техническое задание на информационную систему.

В соответствии с ГОСТ 34.602-89, техническое задание (ТЗ) – основной документ, определяющий требования и порядок создания (развития или модернизации) автоматизированной системы, в соответствии с которым проводится разработка ИС и ее приемка при вводе в действие.

Разработка технического задания предусматривает описание следующих разделов:

- общие сведения; назначение и цели создания (развития) системы;

- характеристика объектов автоматизации;

- требования к системе;

- состав и содержание работ по созданию системы;

- порядок контроля и приемки системы;

- требования к составу и содержанию работ по подготовке объекта автоматизации к вводу системы в действие;

- требования к документированию;

- источники разработки.

Проектная стадия главным образом ориентирована на разработку технического и рабочего проектов. Процесс разработки технического задания включает обследование объекта автоматизации (организации или подразделения) и его систем управления. Для решения задач информационного обеспечения необходимо проанализировать информационные потоки, формы документации, системы кодирования, а

также все связанное со структурой баз данных и СУБД, что определяет состав исходных технологических требований.

Стадия 4. Эскизный проект.

- разработка предварительных проектных решений по системе и ее частям;
- разработка эскизной документации на ИС и ее части.

Если для ИС конкретного объекта автоматизации проектные решения выбраны ранее или являются очевидными, стадия эскизного проекта может быть исключена из последовательности работ. Таким образом, эта стадия не является строго обязательной.

На этапе эскизного проекта, в том числе, должны быть определены:

- цели, функции ИС и подсистем;
- состав комплексов задач и отдельных задач;
- концепция и структура информационной базы;
- функции СУБД;
- функции и параметры основных программных средств;
- ожидаемый эффект от ее внедрения.

Документация, содержащая результаты работ по совокупности принятых проектных решений, согласовывается, утверждается и используется в дальнейшем для выполнения работ по созданию ИС.

На основании технического задания (в том числе, при наличии эскизного проекта) разрабатывается технический проект ИС.

Стадия 5. Технический проект.

- разработка проектных решений по системе и ее частям;
- разработка документации на ИС и ее части;
- разработка и оформление документации на поставку изделий для комплектования ИС и (или) технических требований (технических заданий) на их разработку;
- разработка заданий на проектирование в смежных частях проекта объекта автоматизации.

На этапе технического проекта проводятся работы научно-исследовательского и экспериментального характера для выбора основных

проектных решений, а также рассчитывается экономическая эффективность системы.

Важным аспектом разработки технического проекта является анализ всей используемой информации на предмет таких характеристик, как полнота, отсутствие дублирования и избыточности, непротиворечивость и т.д., а также определение форм выходных документов. Документация должна быть оформлена в соответствии с требованиями ГОСТ 34-201 и РД 50-34.698.

Стадия 6. Рабочая документация.

- разработка рабочей документации на систему и ее части;
- разработка или адаптация программ.

Один из основных этапов стадии рабочего проектирования – разработка рабочей документации на информационное обеспечение ИС, в состав которой входят: технический проект ИС, описание баз данных, перечень исходных и выходных данных и документов.

Стадия технического проектирования завершается подготовкой и оформлением документации на поставку для комплектования ИС и определением технических требований и составлением ТЗ на разработку ИС.

Стадия «Рабочая документация» предполагает создание, как программного продукта, так и всей сопровождающей документации, которая должна предоставлять все сведения, обеспечивающие выполнение работ на стадиях ввода ИС в действие и эксплуатации ИС, в том числе, сведения для поддержания уровня качества ИС (соблюдения эксплуатационных характеристик).

Послепроектная стадия включает в себя реализацию мероприятий по внедрению, подготовку помещений и технических средств, обучение персонала. Также производится эксплуатация системы с решением конкретных задач, анализируются результаты испытаний, реализуются мероприятия по сопровождению ИС.

Стадия 7. Ввод в действие.

- подготовка объекта автоматизации к вводу ИС в действие;
- подготовка персонала;

- комплектация ИС поставляемыми изделиями (программными и техническими средствами, программно-техническими комплексами, информационными изделиями);

- строительно-монтажные работы;
- пусконаладочные работы;
- проведение предварительных испытаний;
- проведение опытной эксплуатации;
- проведение приемочных испытаний.

Основными видами испытаний для ИС являются такие, как: предварительные испытания, опытная эксплуатация и приемочные испытания, которые при необходимости могут быть расширены дополнительными испытаниями ИС и ее составляющих частей.

В ходе предварительных испытаний, регламентируемых соответствующей программой и методикой, главным образом проводятся испытания системы на работоспособность и соответствие ТЗ, а также устранение неисправностей и внесение изменений в документацию на ИС.

На следующем этапе происходит процесс проведения опытной эксплуатации, анализируются ее результаты, и при необходимости проводится доработка ПО и дополнительная наладка технических средств ИС.

В процессе проведения приемочных испытаний реализуются испытания на соответствие ТЗ, анализируются результаты испытания системы, устраняются недостатки, которые были выявлены при испытаниях.

При всех видах испытаний оформляются соответствующие акты о приемке системы в опытную эксплуатацию, ее завершении и о приемке системы в постоянную эксплуатацию.

Стадия 8. Сопровождение ИС.

- выполнение работ в соответствии с гарантийными обязательствами;
- послегарантийное обслуживание.

Основными процессами этой стадии являются осуществление работ по устранению недостатков, выявленных при эксплуатации системы в течение гарантийных сроков, а также анализ функционирования системы, выявление

отклонений и их причин, устранение причин отклонений и недостатков, обеспечение стабильности эксплуатационных характеристик.

2.3 Типовое проектирование ИС, типовое проектное решение (ТПР)

Методы типового проектирования направлены на выполнение проектирования ИС с использованием типовых проектных решений.

Типовое проектное решение – проектное решение, пригодное к многократному использованию (тиражируемое проектное решение).

Применение методов типового проектирования имеет свои особенности. Основным условием для использования таких методов является возможность декомпозиции проектируемой ИС на составляющие компоненты (подсистемы, программные модули, комплексы выполняемых задач и т.д.), для реализации которых можно выбрать типовые проектные решения, существующие на рынке, которые будут настроены на нужды конкретного предприятия.

Помимо собственно функциональных (программных, аппаратных) элементов, типовое решение подразумевает наличие необходимой документации, в которой дается детальное описание ТПР (в том числе, процедур настройки), отвечающее требованиям проектируемой системы.

По уровню декомпозиции системы можно выделить такие классы ТПР, как:

- элементные ТПР – ТПР по отдельному элементу (задаче, виду обеспечения);
- подсистемные ТПР – ТПР по отдельным подсистемам;
- объектные ТПР – отраслевые ТПР, включающие весь набор подсистем ИС.

Выделенные классы ТПР имеют свои достоинства и недостатки. Рассмотрим наиболее характерные из них.

К достоинству элементных ТПР можно отнести реализацию модульного подхода к проектированию ИС. В то же время это приводит к

большим затратам на доработку ТПР конкретных элементов и к затратам на объединение разных элементов вследствие их несовместимости.

Подсистемные ТПР также позволяют реализовать модульный подход к проектированию ИС. Кроме того, они позволяют осуществлять параметрическую настройку компонентов на объекты различных уровней управления; взаимосвязанные компоненты и высокая степень интеграции элементов ИС приводят к минимизации затрат на проектирование и программирование. Однако в случае нескольких производителей программного обеспечения появляются проблемы в объединении различных функциональных подсистем; помимо этого, с точки зрения непрерывного реинжиниринга процессов адаптивность ТПР является недостаточной.

Объектные ТПР имеют такие преимущества, как:

- масштабируемость (допускаются конфигурации ИС для различного числа рабочих мест);
- методологическое единство компонентов ИС;
- совместимость компонентов ИС;
- открытость архитектуры (возможность развертывания ТПР на платформах разного типа);
- конфигурируемость (возможность использовать необходимое подмножество компонентов системы).

К недостаткам объектных ТПР можно отнести проблемы реализации типового проекта в оригинальном объекте управления, что приводит в определенных ситуациях к необходимости смены организационной структуры объекта автоматизации.

При реализации типового проектирования применяются такие подходы, как: **параметрически-ориентированное** и **модельно-ориентированное** проектирование.

Этапами **параметрически-ориентированного** проектирования являются:

- постановка задач и определение пригодности пакетов прикладных программ (ППП) для их решения через систему критериев оценки;

- анализ доступных ППП исходя из критериев; выбор и приобретение подходящего ППП;

- настройка параметров приобретенного ППП.

Среди критериев оценки ППП выделяют [20] следующие группы:

- назначение и возможности пакета;
- отличительные признаки и свойства пакета;
- требования к техническим и программным средствам;
- документация пакета; факторы финансового порядка;
- особенности установки пакета;
- особенности эксплуатации пакета;
- помощь поставщика по внедрению и поддержанию пакета;
- оценка качества пакета и опыт его использования;
- перспективы развития пакета.

Отметим, что каждая из перечисленных групп критериев может быть детализирована на совокупность частных показателей, дающих дополнительную информацию для каждого аспекта анализа, выбранного ППП. Значения критериев определяются с использованием методов экспертного оценивания.

Другим подходом реализации типового проектирования является модельно-ориентированное проектирование, сущность которого состоит в адаптации существующих характеристик типовой ИС, исходя из модели объекта автоматизации, построение которой предполагает использование специального программного инструментария.

При таком подходе технология проектирования должна иметь средства как для работы с моделью конкретного предприятия, так и с моделью типовой ИС.

В репозитории типовой ИС содержится модель объекта автоматизации, которая является основой для конфигурирования программного обеспечения. Кроме того, в репозитории содержится базовая (ссылочная) модель ИС и типовая (референтная) модели ее определенных классов.

Базовая модель ИС описывает бизнес-процессы, организационную структуру, бизнес-объекты, бизнес- функции, для поддержки которых предназначены программные модули типовой ИС.

Типовые модели предназначены для описания конфигурации ИС для тех или иных отраслей, типов производства.

Модель конкретного предприятия может быть построена либо в результате выбора фрагментов типовой модели с учетом особенностей объекта автоматизации (BAAN Enterprise Modeler), либо с использованием автоматизированной адаптации этих модулей с учетом мнений экспертов (SAP Business Engineering Workbench). Модель предприятия, на основе которой осуществляется автоматическое конфигурирование и настройка ИС, хранится в репозитории и может быть откорректирована в случае необходимости.

Внедрение типовой ИС начинается с анализа результатов предпроектного обследования предприятия, сформированных в виде требований к конкретной ИС, для оценки которых может быть использована методика оценки ППП. На следующем этапе необходимо построить предварительную модель ИС, которая должна полно отражать особенности реализации ИС для конкретного объекта автоматизации. Предварительная модель – основа для выбора типовой модели системы, а также для формирования перечня компонентов, для реализации которых потребуются другие программные средства или инструментальные средства, имеющиеся в составе типовой ИС.

При реализации типового проекта имеет место выполнение следующих операций [20]:

- установку глобальных параметров системы;
- задание структуры объекта автоматизации;
- определение структуры основных данных;
- задание перечня реализуемых функций и процессов; описание интерфейсов;
- описание отчетов;
- настройку авторизации доступа;

- настройку системы архивирования.

Типовое проектирование в настоящее время широко представлено в современных средствах.

3 Практикум по проектированию информационных систем

Лабораторная работа № 1. Основы работы в BPwin 4.1

С помощью этой лабораторной работы Вы сможете:

- познакомиться с основными элементами рабочего интерфейса BPwin 4.1;
- освоить технологию создания новой модели;
- научиться редактировать построенную диаграмму.

Теоретические сведения

📖 Для упрощения процесса построения модели бизнес-процессов в IDEF0, можно использовать CASE-средство BPwin, которое позволяет наглядно представить функциональную декомпозицию системы.

AllFusion Process Modeler 4.1 (BPwin) является инструментальным средством, полностью поддерживающим стандарт IDEF0, который был принят в начале 90-х годов в США на основе SADT.

Основная идея методологии SADT - построение **древовидной функциональной модели** предприятия.

Сначала функциональность предприятия описывается в целом, без подробностей. Такое описание называется **контекстной диаграммой** (рис. 6).

Взаимодействие с окружающим миром описывается в терминах **входа** (данные или объекты, потребляемые или изменяемые функцией), **выхода** (основной результат деятельности функции, конечный продукт), **управления** (стратегии и процедуры, которыми руководствуется функция) и **механизмов** (необходимые ресурсы).

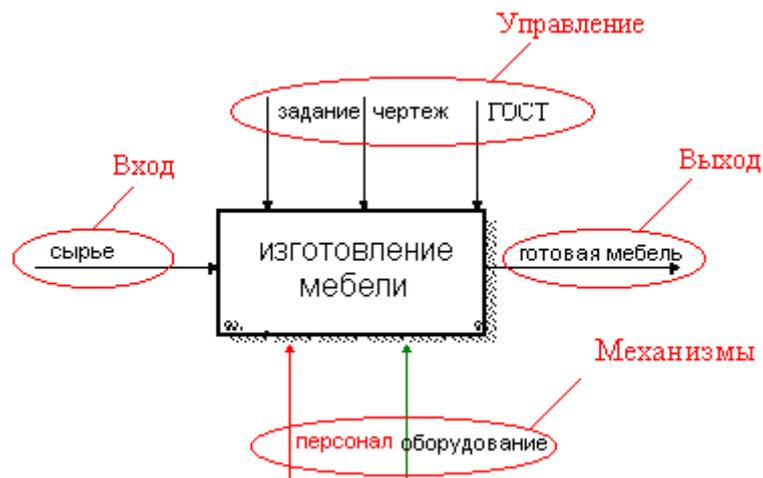


Рисунок 6 - Пример диаграммы IDEF0

Кроме того, при создании контекстной диаграммы формулируются *цель моделирования, область* (описания того, что будет рассматриваться как компонент системы, а что как внешнее воздействие) и *точка зрения* (позиция, с которой будет строиться модель). Обычно в качестве точки зрения выбирается точка зрения лица или объекта, ответственных за работу моделируемой системы в целом.

Общая функция разбивается на крупные подфункции. Этот процесс называется ***функциональной декомпозицией***.

Затем каждая подфункция декомпозируется на более мелкие - и так далее до достижения необходимой детализации описания. Таким образом, формируется диаграмма IDEF0.

Этапы построения модели

1. Определение основного бизнес-процесса.
2. Построение контекстной диаграммы.
3. Построение диаграммы процесса верхнего уровня.
4. Функциональная декомпозиция каждого процесса, с помощью детализирующих диаграмм.

Практическое задание

Создание диаграммы А-0

Построение модели бизнес-процесса рассмотрим на примере работы мебельной фабрики. Во время проведения обследования предприятия были

выявлены её целевые задачи, функциональные деятельности каждого из подразделений предприятия и функциональные взаимодействия между ними; информационные потоки внутри подразделений и между ними; внешние по отношению к предприятию объекты и внешние информационные воздействия, а так же нормативно-справочная документация, данные по имеющимся на предприятии средствам и системам автоматизации.

Целевые функции мебельной фабрики:

- переработка сырья;
- изготовление деталей для мебели;
- сборка изделия;
- контроль качества.

Нормативные документы мебельной фабрики:

- чертежи (деталей, сборочный);
- нормы по переработке сырья;
- стандарты качества;
- производственные инструкции;
- инструкции по технике безопасности.

Подразделения предприятия:

- цех по обработке сырья и бракованных изделий;
- цех по изготовлению деталей;
- сборочный цех;
- отдел проверки качества изделия.

Основным сырьем для изготовления мебели является *дерево*.

Определим основной бизнес-процесс, используя выявленные целевые функции. Так как основное предназначение мебельной фабрики состоит в том, чтобы изготавливать мебель, значит, основным бизнес-процессом является **ИЗГОТОВЛЕНИЕ МЕБЕЛИ**.

Создадим контекстную диаграмму. В данном случае она представляет собой самое общее описание системы. Напомню, что в модели может быть только одна контекстная диаграмма.

Для построения контекстной диаграммы нам необходимо определить входную информацию (данные или материальные ресурсы), которая преобразуется в процессе для получения результата; выходную информацию - готовый результат; управление, которое влияет на процесс, но не преобразуется процессом.; механизмы, которые выполняют процесс.

Для контекстного процесса ИЗГОТОВЛЕНИЕ МЕБЕЛИ определим необходимую информацию:

- ВХОД - сырьё;
- УПРАВЛЕНИЕ – чертежи, производственные инструкции, инструкции по технике безопасности (нормативные документы);
- МЕХАНИЗМЫ – персонал, производственное оборудование;
- ВЫХОД – готовая мебель.



Задание 1. Запуск программы и открытие проекта. Создание новой модели.

1. Запустите **BPwin**.
2. Выберите режим работы: создать новую модель (**Create model**).
3. В поле **Name** введите имя модели: **Изготовление мебели**.
4. Из группы **Type** выберите тип диаграммы **Business Process (IDEF0)** (**IDEF0**) (рис. 7).

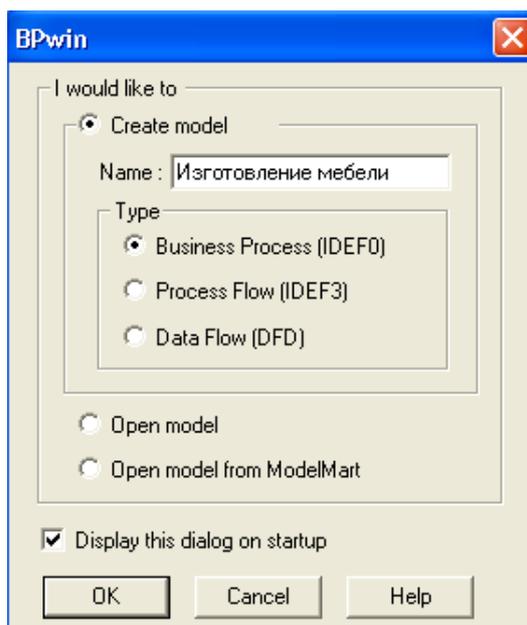


Рисунок 7 - Диалоговое окно создания модели

5. Нажмите ОК.

После щелчка на кнопке ОК появляется диалог **Properties for New Models** (рис. 8).

6. Укажите свою фамилию и инициалы.

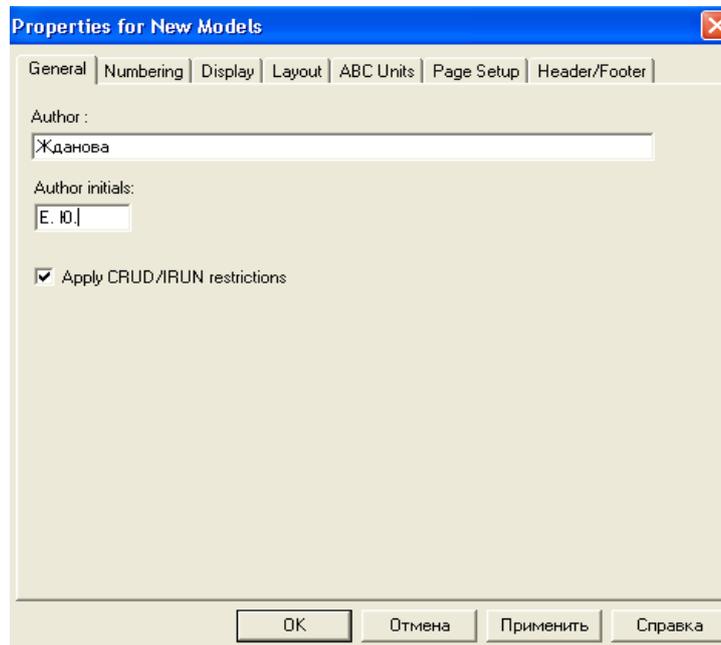


Рисунок 8 - Диалоговое окно Properties for New Models

После этого вы увидите стандартное окно VPwin (рис. 9).

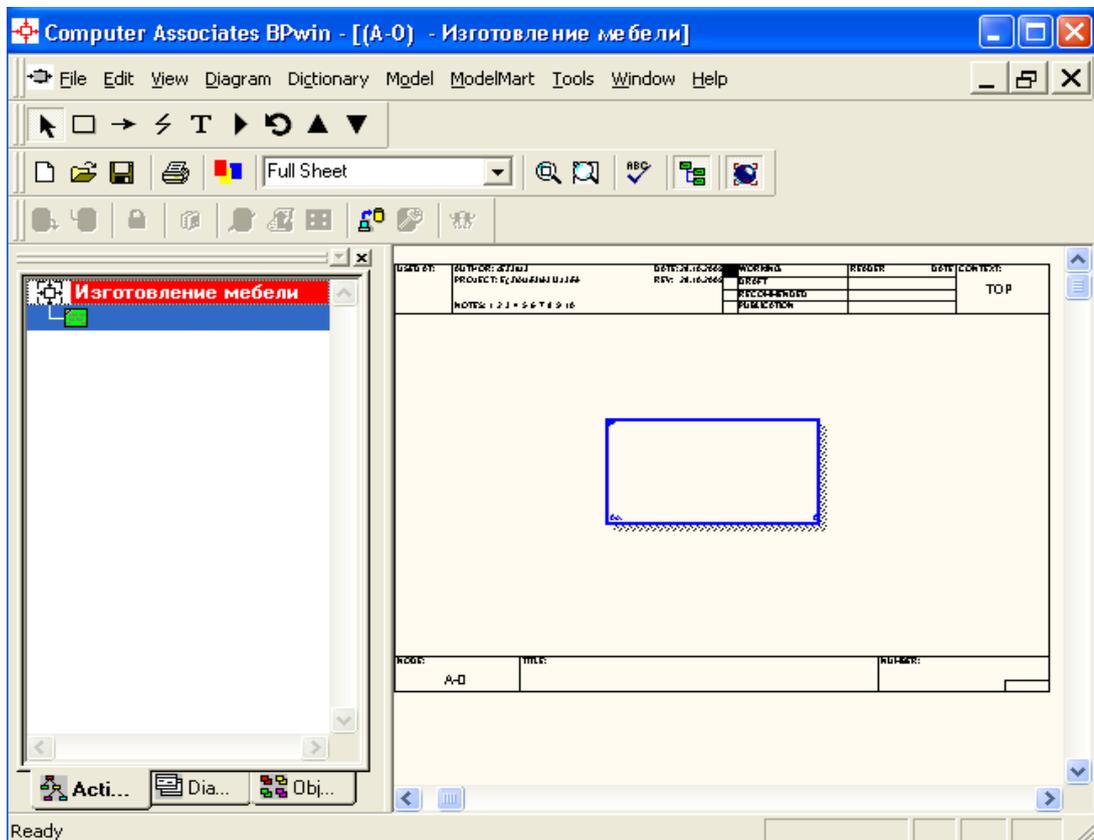


Рисунок 9 - Стандартное окно VPwin

Основные инструменты VPwin

Все основные действия с диаграммами, такие как создание, редактирование и т.д., можно выполнить с помощью главного меню, либо контекстно-зависимого меню (меню, появляющееся при нажатии правой кнопки мыши). Принципы работы с меню являются стандартными для среды Windows: объект сначала делается активным, затем над ним осуществляются необходимые действия.

На основной панели инструментов расположены элементы управления, в основном знакомые по другим Windows-интерфейсам (рис. 10).



Рисунок 10 - Элементы управления VPwin

Функциональность панели инструментов доступна из основного меню VPwin (табл. 1).

Таблица 1 - Элементы управления основной панели инструментов VPwin

Элемент управления	Описание	Соответствующий пункт меню
	Создать новую модель	File/New
	Открыть модель	File/Open
	Сохранить модель	File/Save
	Напечатать модель	File/Print
	Вызвать генератор отчетов Report Builder	Tools/Report Builder
	Выбор масштаба	View/Zoom
	Масштабирование	View/Zoom
	Проверка правописания	Tools/Spelling
	Включение и выключение навигатора модели Model Explorer	View/Model Explorer
	Включение и выключение дополнительной панели инструментов работы с ModelMart	ModelMart

На основной панели инструментов (либо в любом желаемом месте экрана) расположены инструменты редактора BPwin для IDEF0-диаграмм (рис. 11).

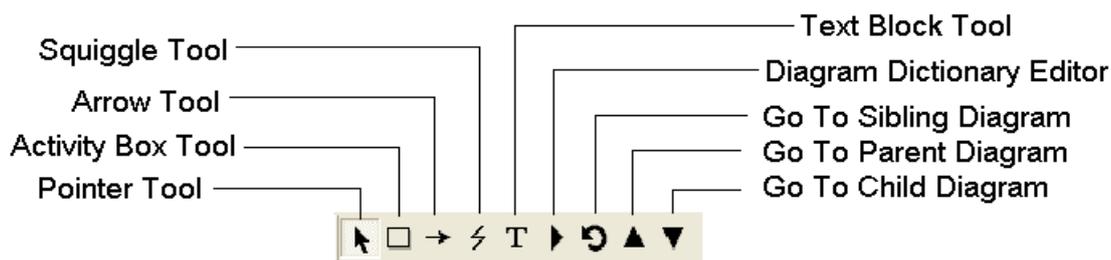


Рисунок 11 - Панель инструментов редактора BPwin

Таблица 2 - Инструменты редактора BPwin

Элемент редактирования	Описание	Название элемента
	Выбор и определение позиции объектов, добавленных в диаграмму.	Pointer Tool
	Установка блоков в диаграмме.	Activity Box Tool
	Установка дуги в диаграмме.	Arrow Tool
	Создание тильды (squiggle), которая соединяет дугу с ее названием.	Squiggle Tool
	Создание текстовых блоков.	Text Block Tool
	Отображение следующей диаграммы того же уровня	Go to Sibling Diagram
	Открытие диалогового окна Diagram Dictionary Editor, где можно перейти на какую-либо диаграмму или создать новую диаграмму.	Diagram Dictionary Editor
	Переход на родительскую диаграмму.	Go to Parent Diagram
	Отображение диаграммы потомка или разложение выделенного блока на диаграмму потомка	Go to Child Diagram



Задание 2. Работа с блоком. Построение контекстной диаграммы процесса ИЗГОТОВЛЕНИЕ МЕБЕЛИ.

Для ввода имени блока необходимо:

1. Щелкнуть правой клавишей мыши по блоку.
2. Выбрать команду **Name**.

3. В диалоговом окне ввести название «Изготовление мебели» (рис.12).

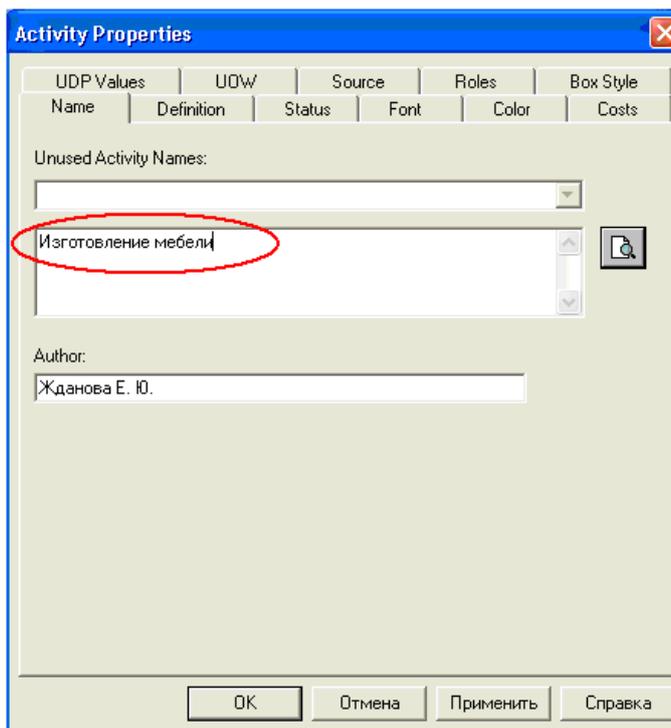


Рисунок 12 - Диалоговое окно VРwin

4. Для того чтобы текст стал понятен, в контекстном меню выберите пункт **Font** (рис. 13).

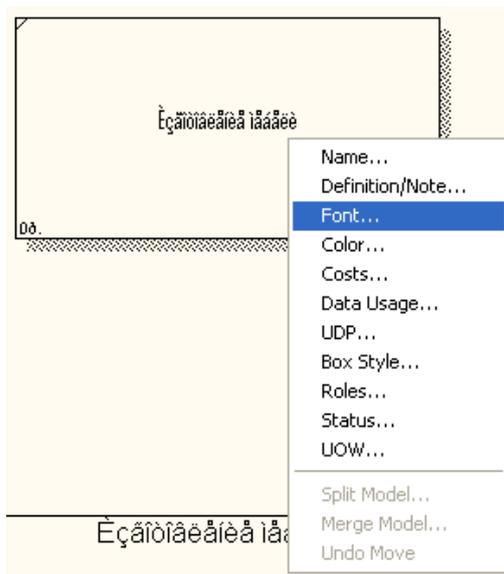


Рисунок 13 - Контекстно-зависимое меню

5. В диалоговом окне **Activity Properties** в нижней части вкладки **Font** установите флажки в опциях **Apply setting to**, позволяющих изменить шрифт для всех работ на текущей диаграмме, в модели, и в группе **Global**,

позволяющей изменить шрифт одновременно для всех объектов модели, в опции **Script** выберите «**кириллический**».

6. Установите шрифт **Arial Unicode MS**, курсив, 16 пт (рис. 14).

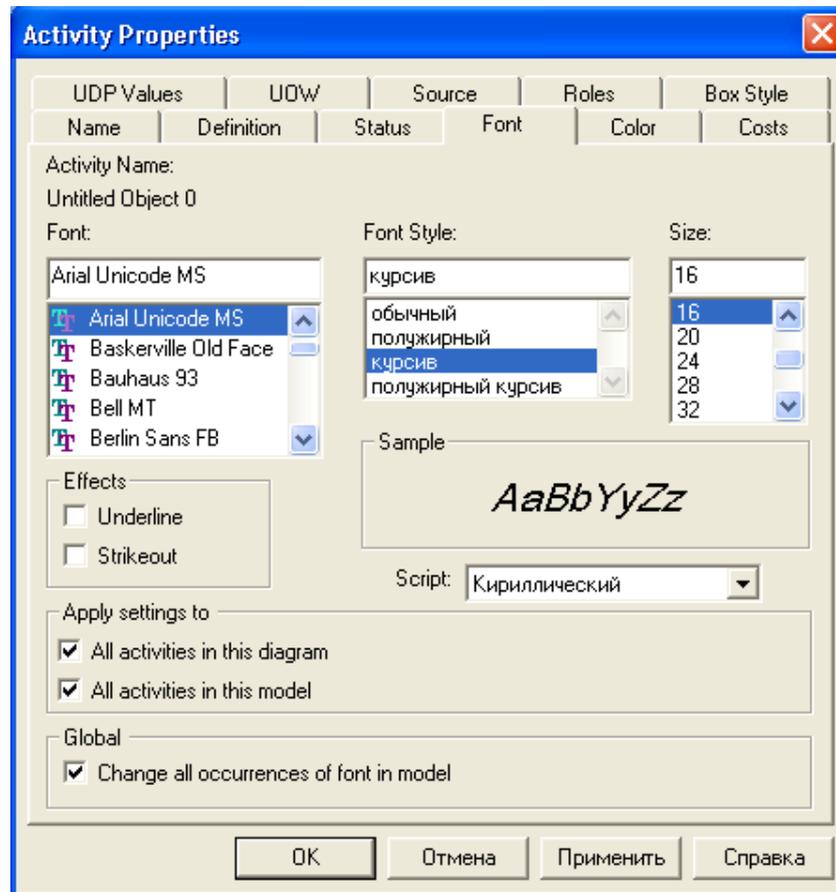


Рисунок 14 - Вкладка Font диалога Activity Properties

 После того, как эта работа будет проделана, пригласите преподавателя и продемонстрируйте ему результат.



Задание 3. Построение дуг.

Для построения дуги **управления** необходимо:

1. Выбрать на панели инструментов кнопку .
2. Подвести курсор к верхнему краю окна построения диаграммы до появления черной полосы и произвести щелчок левой кнопкой мыши по этой полосе (рис. 15).



Рисунок 15 - Начало построения дуги

3. Подвести курсор мыши к верхней стороне блока до образования темного треугольника и щелкнуть левой кнопкой мыши (рис. 16).

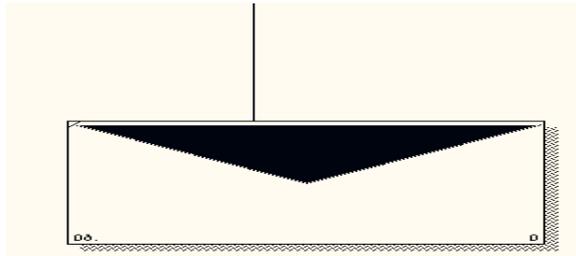


Рисунок 16 - Построение дуги управления

Построение дуг **входа** и **механизмов** производится аналогичным образом.

4. Постройте дуги **входа** и **механизмов**.

Для построения дуги **выхода** выполняются те же действия, но в противоположном порядке: от правой стороны блока к правой стороне окна построения диаграмм.

5. Постройте дугу **выхода**.



Задание 4. Идентификация дуги управления.

1. Выберите на панели редактирования кнопку .
2. Щелкните правой кнопкой мыши по дуге.
3. Выберите команду **Name** (рис. 17).

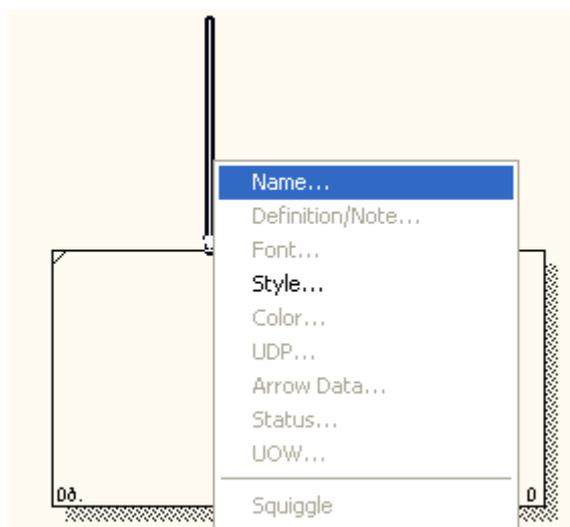


Рисунок 17 - Контекстно-зависимое меню

4. В диалоговом окне введите название дуги: «Нормативная документация» (рис. 18).

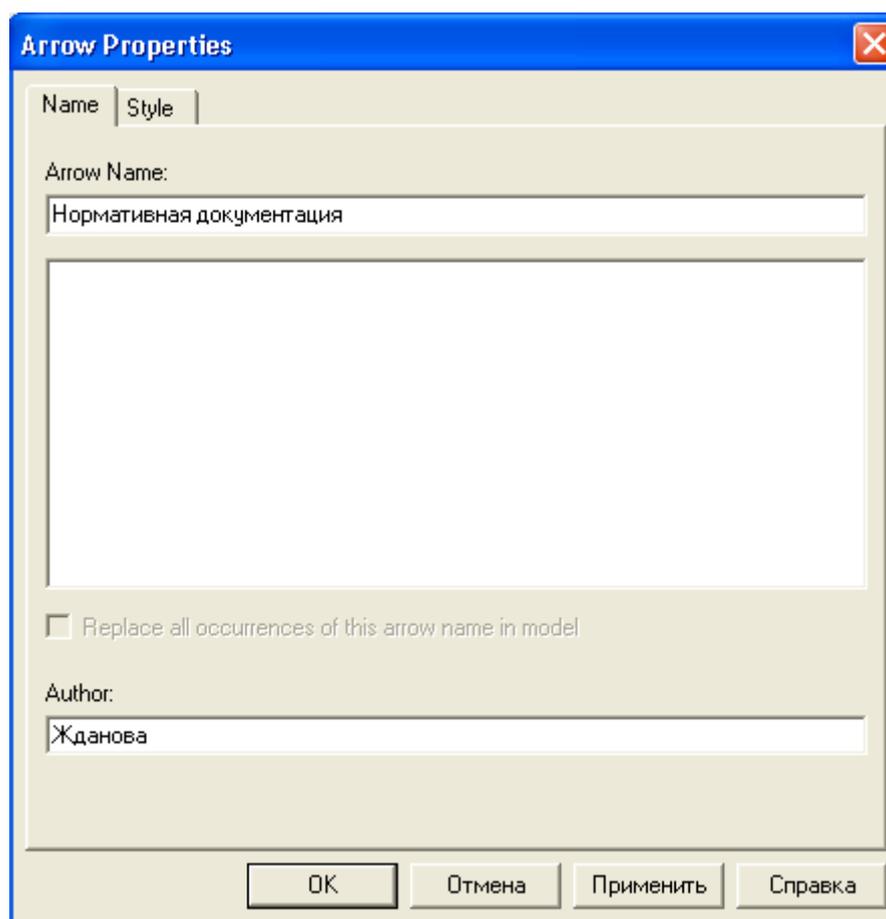


Рисунок 18 - Диалоговое окно Arrow Properties

5. Для того чтобы текст для дуги стал понятен (рис 19), выберите меню **Model - Default Fonts**.

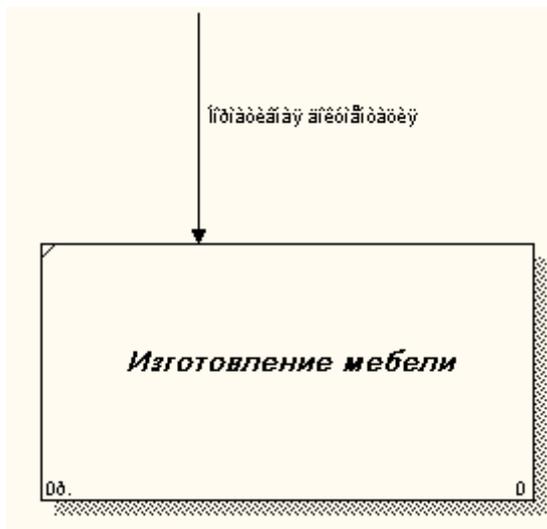


Рисунок 19 - Фрагмент диаграммы

BPwin позволяет установить шрифт по умолчанию для объектов определенного типа (например, дуг) на диаграммах и в отчетах. После выбора меню **Model - Default Fonts** появляется каскадное меню, каждый пункт которого служит для установки шрифтов для определенного типа объектов (рис. 20).

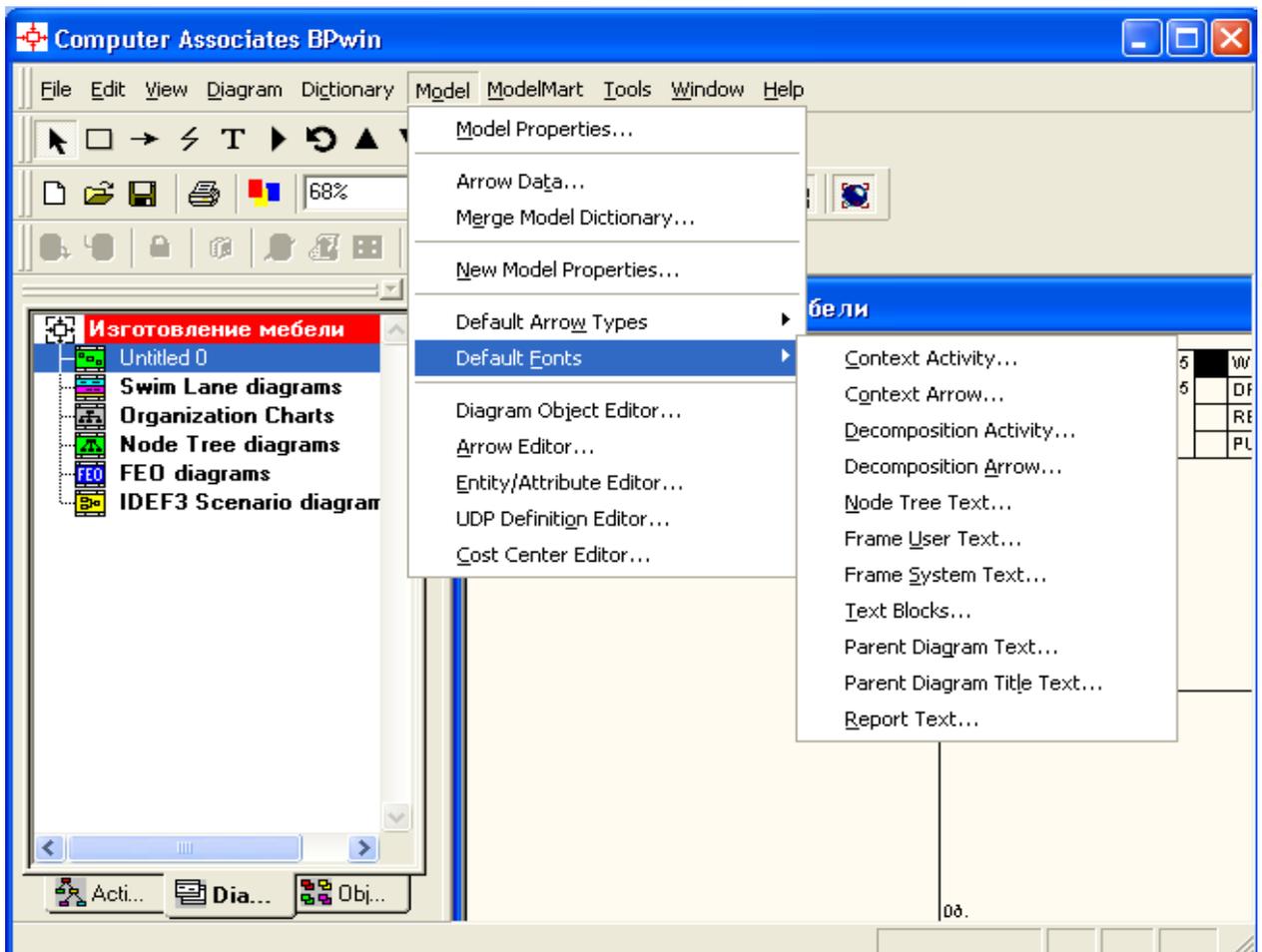


Рисунок 20 - Диалог Model - Default Fonts

1. *Context Activity* - работа на контекстной диаграмме;
2. *Context Arrow* - стрелки на контекстной диаграмме;
3. *Decomposition Activity* - работы на диаграмме декомпозиции;
4. *Decomposition Arrow* - стрелки на диаграмме декомпозиции;
5. *Node Tree Text* - текст на диаграмме дерева узлов;
6. *Frame User Text* - текст, вносимый пользователем в каркасе диаграмм;
7. *Frame System Text* - системный текст в каркасе диаграмм;
8. *Text Blocks* - текстовые блоки;
9. *Parent Diagram Text* - текст родительской диаграммы;
10. *Parent Diagram Title Text* - текст заголовка родительской диаграммы;
11. *Report Text* - текст отчетов.

6. В диалоговом окне **Default Context Arrow Name Text Font** в нижней части установите флажок в опции **Change all occurrences**, позволяющей изменить шрифт для названий всех дуг на текущей диаграмме, в опции **Script** выберите «**кириллический**».

7. Установите шрифт **Arial Unicode MS**, курсив, 14 пт. (рис. 21).

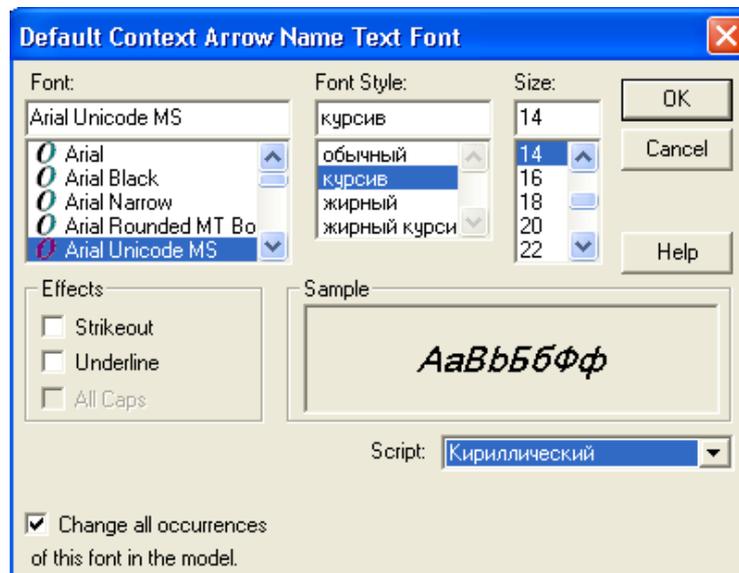


Рисунок 21 - Диалоговое окно Default Context Arrow Name Text Font

Вот что должно у вас получиться (рис. 22).

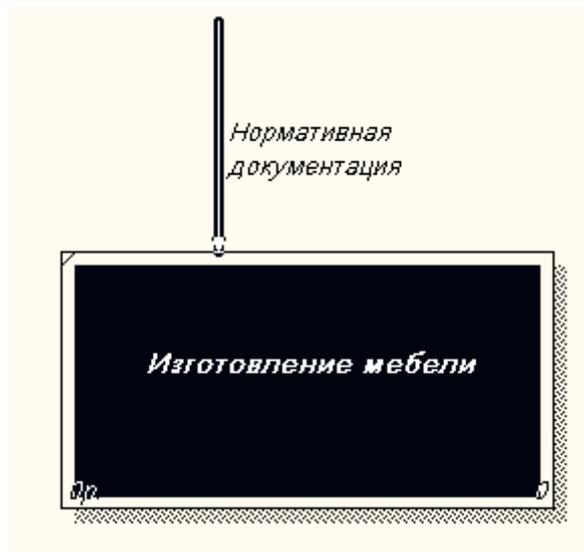


Рисунок 22 - Фрагмент диаграммы



Задание 5. Работа с блоком.

Самостоятельно постройте дуги:

- ВХОДА: «Сырьё»;
- МЕХАНИЗМА: «Персонал», «Производственное оборудование»;
- ВЫХОДА: «Готовая мебель».

 После того, как эта работа будет проделана, пригласите преподавателя и продемонстрируйте ему результат.



Задание 6. Установление тильды.

Название дуги является независимым объектом, который можно перемещать относительно дуги. Текст может располагаться по отношению к дуге в свободной форме, либо соединяться с дугой символом тильды.

Чтобы установить тильду следует:

1. На панели инструментов нажать кнопку  ;
2. Щелкнуть левой кнопкой мыши по тексту, а затем по дуге (рис. 23);

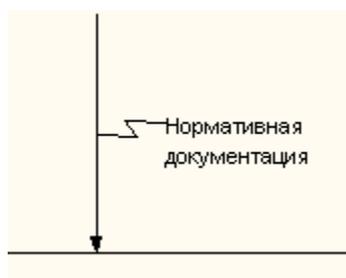


Рисунок 23 - Установление тильды

3. Можно также использовать команду контекстно-зависимого меню **Squiggle** (рис. 24).

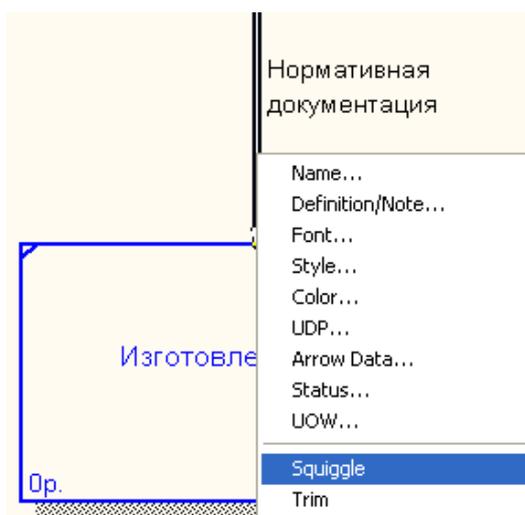


Рисунок 24 - Контекстно-зависимое меню

Дуга представляет собой совокупность отдельных графических объектов: прямые участки, изогнутые участки, изображение наконечника. Отдельные элементы можно передвигать независимо друг от друга, меняя форму дуги, также дугу можно перемещать как единый неделимый элемент.

4. Установите тильду к остальным дугам и их названиям.



Задание 7. Изменение цвета текста, фона блока, цвета и стиля дуг.

1. Для изменения цвета текста выполните команду контекстно-зависимого меню **Color** (рис. 25).

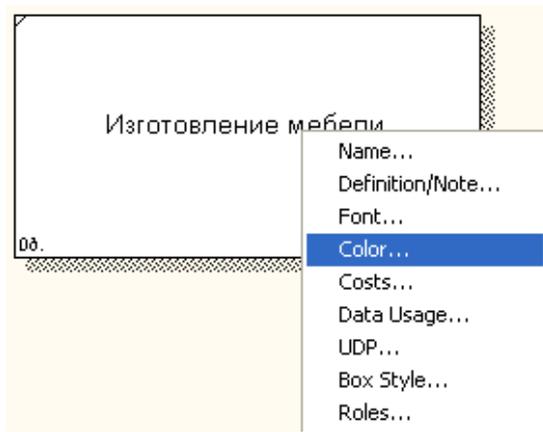


Рисунок 25 Контекстно-зависимое меню

2. Выберите цвет и нажмите кнопку  (рис. 26).

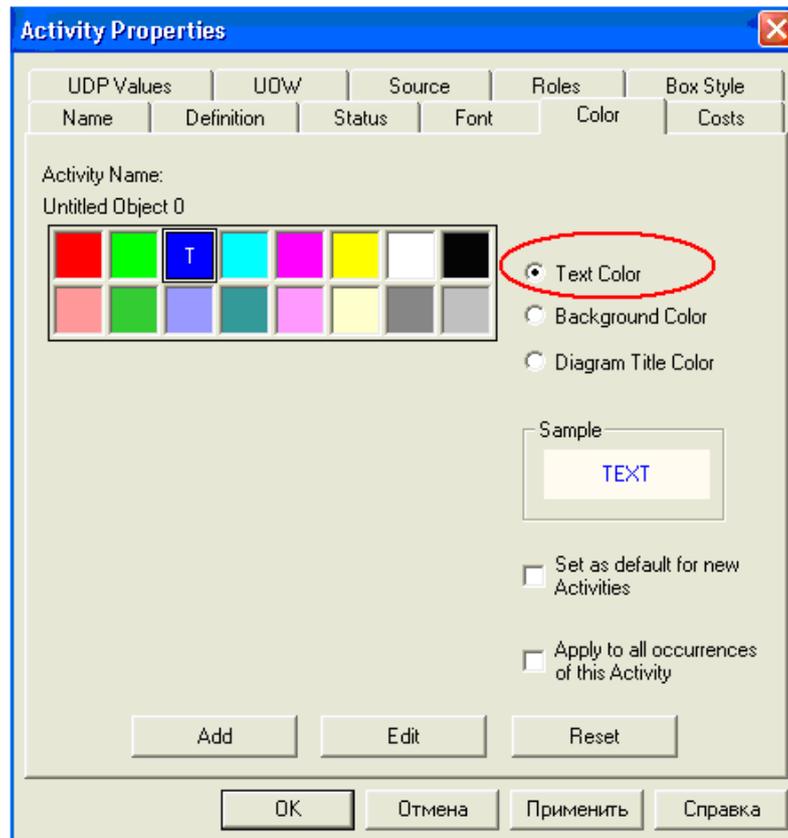


Рисунок 26 - Диалоговое окно выбора цвета текста и стрелок

3. Для изменения фона блока выберите **Background Color** и цвет (рис. 27).

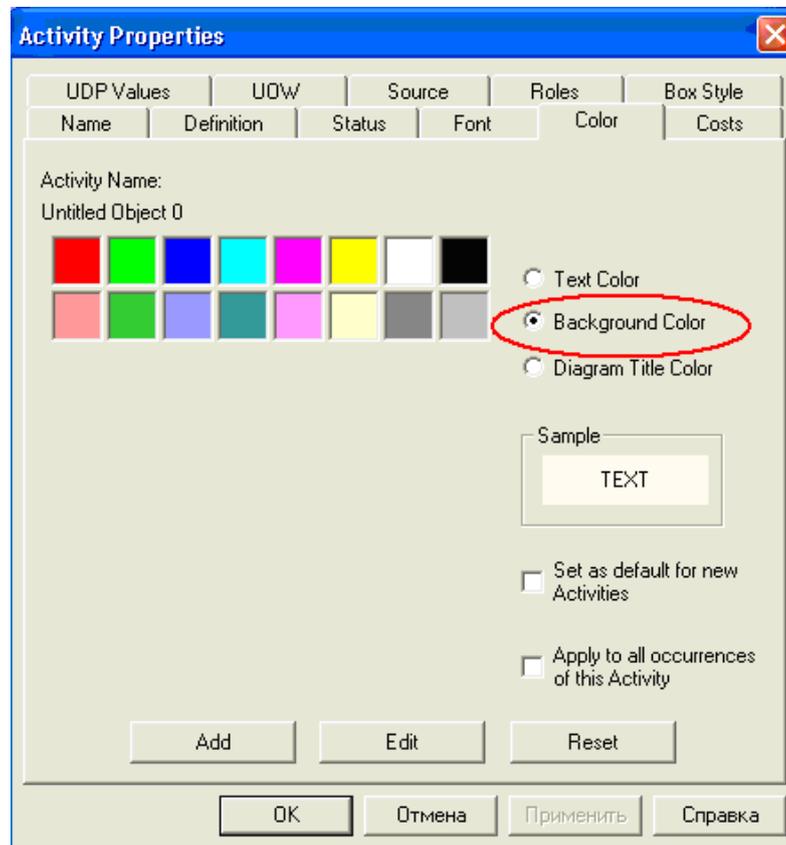


Рисунок 27 - Вкладка Color диалога Activity Properties

4. Для изменения стиля дуги выберите в контекстно-зависимом меню команду **Style** (рис. 28):

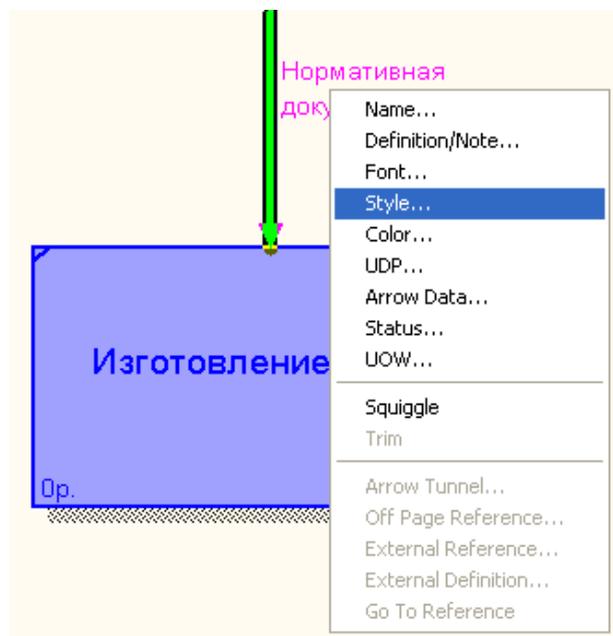


Рисунок 28 - Контекстно-зависимое меню

5. В диалоговом окне укажите тип и стиль дуги, нажмите на кнопку ОК (рис. 29).

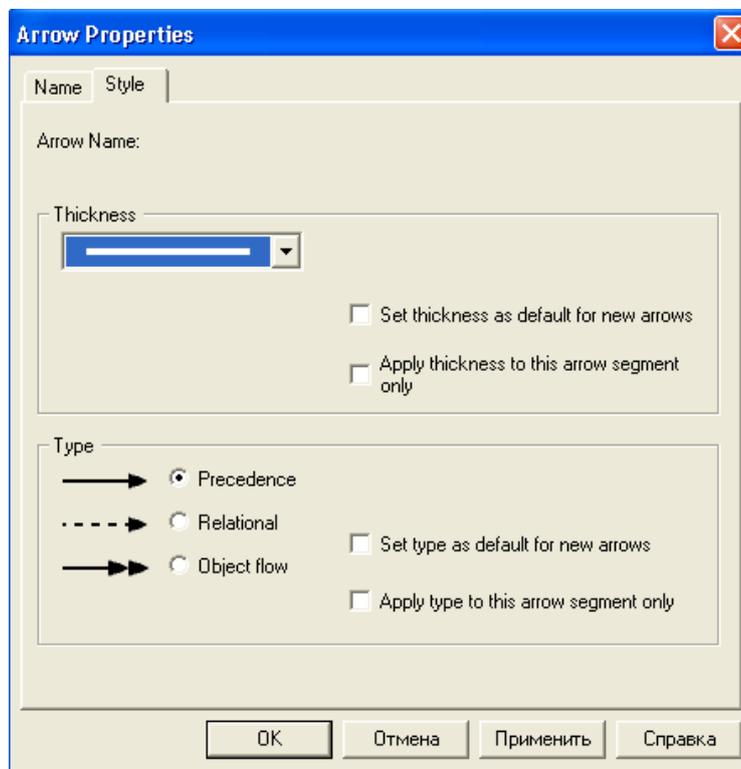


Рисунок 29 - Диалоговое окно Arrow Properties

Вот что должно у вас получиться (рис. 30).

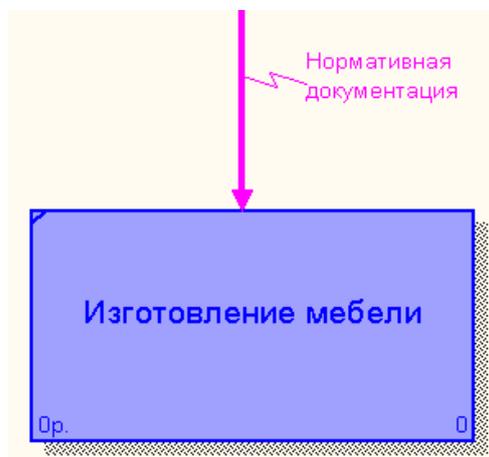


Рисунок 30 - Пример форматирования элементов диаграммы

Удаление блоков, дуг или текста.

Для удаления блока и дуги или текста необходимо их активизировать щелчком левой кнопки мыши и нажать клавишу Delete, а затем подтвердить намерения по поводу удаления.



Задание 8. Форматирование диаграммы.

1. Самостоятельно произведите форматирование всех элементов диаграммы, опираясь на данные, приведенные в таблице 3.

Таблица 3 - Таблица редактирования

Объект	Действие	Порядок выполнения
Блок	Изменение размера	Для изменения высоты перетащить мышью верхнюю или нижнюю границу блока, аналогично меняется размер по горизонтали
Текст	Стандартные способы форматирования	Правой клавишей мышки щелкнуть по тексту, выбрать пункт контекстно-зависимого меню Font Editor .
Дуга	Изменение стиля, цвета, размера	Щелкнуть правой клавишей мышки по дуге и выбрать соответствующий пункт: Style Editor, Color Editor или Trim
Удаление блоков, дуг или текста: активизировать щелчком левой кнопки мышки необходимый объект и нажать Delete , а затем подтвердить запрос на удаление.		

После выполнения задания у вас должна получиться следующая контекстная диаграмма (рис. 31).

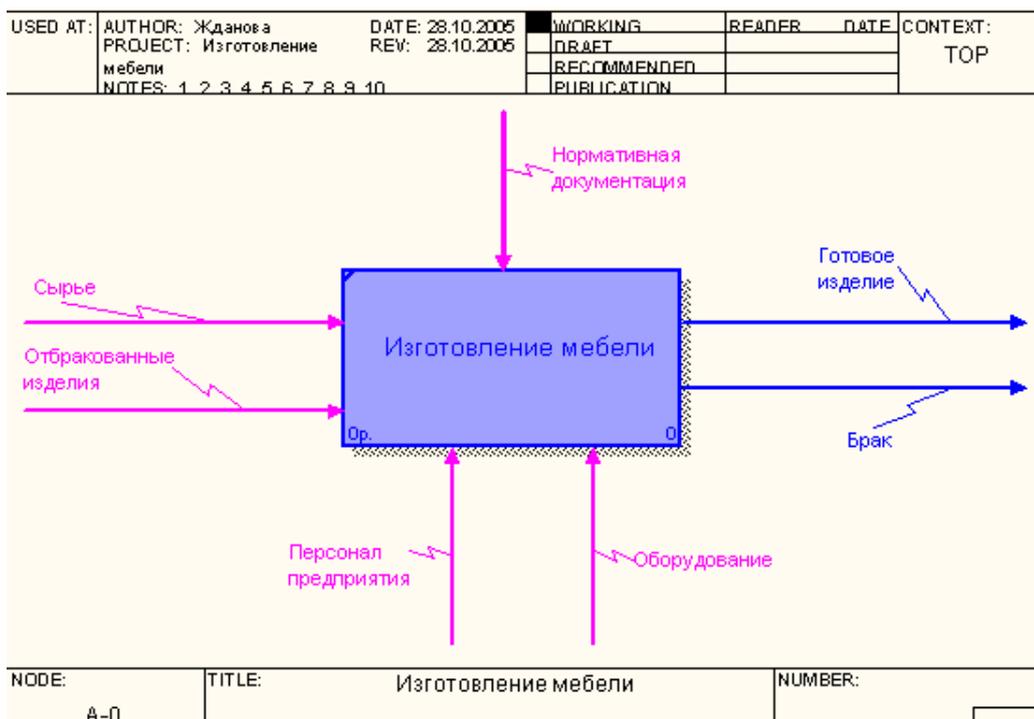


Рисунок 31 - Контекстная диаграмма процесса «Изготовление мебели»

 После того, как эта работа будет проделана, пригласите преподавателя и продемонстрируйте ему результат.



Задание 9. Сохранение полученной диаграммы.

Сохраните полученную диаграмму.

1. Создайте папку, назовите её своей фамилией и в неё сохраняйте свои работы.
2. В меню **File** выберите команду **Save as**.
3. Укажите путь к своей папке и имя файла **Lab1.bp1** (рис. 32).

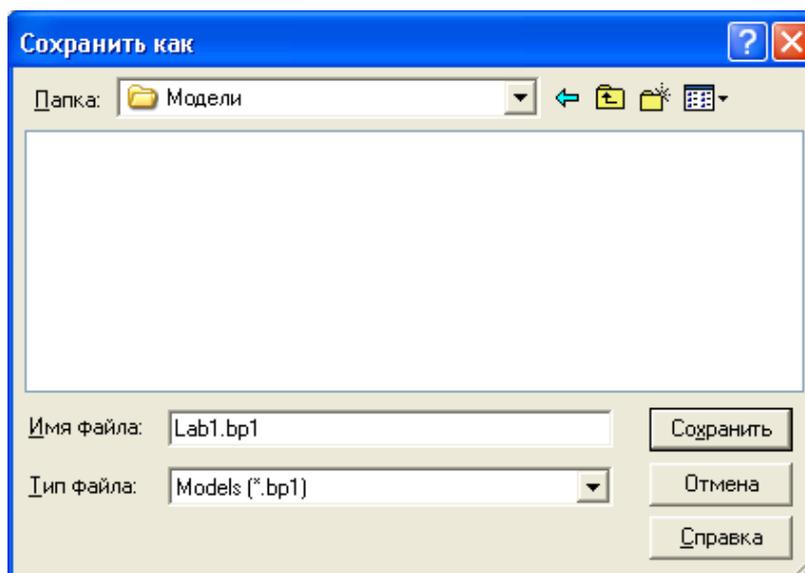


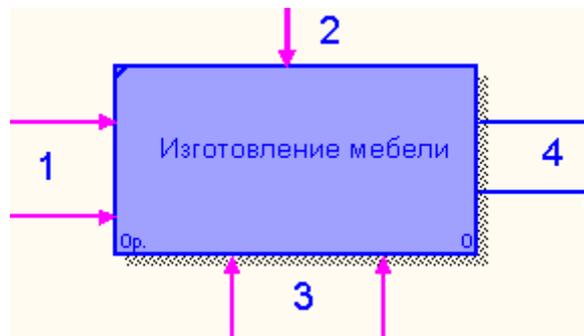
Рисунок 32 - Диалоговое окно сохранения документа

4. Нажмите на кнопку **СОХРАНИТЬ**.

Контрольные вопросы

Ответьте на контрольные вопросы:

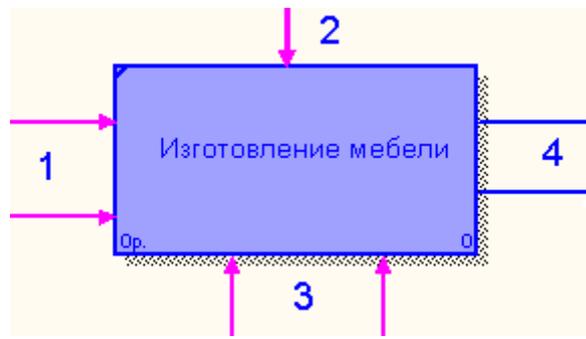
1. Перечислите основные возможности VRwin.
2. Охарактеризуйте основные элементы рабочего интерфейса VRwin.
3. Какую методологию поддерживает VRwin?
4. Укажите назначение каждой из дуг изображенных на рисунке.



5. Назовите основные этапы построения модели.
6. Какой процесс можно назвать функциональной декомпозицией?
7. Перечислите элементы контекстной диаграммы.
8. При помощи какого инструмента строятся дуги на диаграмме?

💡 Тестовое задание

1. Что представляет собой модель бизнес-процессов:
 - a) комплекс диаграмм, каждая из которых описывает отдельный бизнес-процесс;
 - b) иерархию диаграмм, каждая из которых описывает отдельный бизнес-процесс;
 - c) случайный набор диаграмм, каждая из которых описывает отдельный бизнес-процесс;
 - d) все ответы правильные;
 - e) правильного ответа нет.
2. Что входит в состав диаграмм:
 - a) блоки, описывающие подпроцессы (функции системы), и дуги, связывающие блоки вместе и изображающие взаимодействия и взаимосвязи между блоками;
 - b) блоки, описывающие подпроцессы (функции системы);
 - c) дуги, связывающие блоки вместе и изображающие взаимодействия, и взаимосвязи между блоками;
 - d) нумерация диаграммы;
 - e) правильные ответы b), d).
3. Укажите этап, в котором происходит построение диаграммы процесса верхнего уровня:
 - a) рисование дуги управления;
 - b) определение основного бизнес-процесса;
 - c) построение контекстной диаграммы;
 - d) функциональная декомпозиция каждого процесса, с помощью детализирующих диаграмм;
 - e) все ответы правильные.
4. Укажите, для чего предназначена левая сторона блока (1):



- a) для **управления** (правила, стратегии, стандарты);
- b) для **механизмов** (ресурсы, которые выполняют процесс);
- c) для **выходов** (материал или информация, получаемые в результате выполнения процесса);
- d) для **входов** (материал или информация, которые используются или преобразуются для получения результата);
- e) правильные ответы c), d).

5. Какая команда из контекстно-зависимого меню позволит изменить стиль дуги:

- a) Font Editor;
- b) Color Editor;
- c) Trim;
- d) Style Edition;
- e) Background Color;
- f) Squiggle.

6. Для чего предназначен инструмент **T** :

- a) используется для создания тильды (squiggle), которая соединяет дугу с ее названием;
- b) используется для выбора и определения позиции объектов, добавленных в диаграмму;
- c) используется для создания текстовых блоков;
- d) используется для выбора цвета текста и фона;
- e) правильного ответа нет.

👤 После того, как эта работа будет проделана, пригласите преподавателя и продемонстрируйте ему результат.



Контрольные задания

Предложенные задания предназначены для самостоятельной разработки модели информационной системы. Номер варианта задания для каждого учащегося учитель назначает сам. В ходе выполнения задания учащиеся должны составить модель информационной системы для автоматизации рабочего места участника процесса:

- 1) описать главную цель, его функцию;
- 2) определить основной бизнес-процесс;

- 3) построить контекстную диаграмму;
- 4) построить диаграммы верхнего уровня;
- 5) произвести функциональную декомпозицию каждого процесса с помощью детализирующих диаграмм;
- 6) составить отчет;
- 7) построить DFD - диаграмму;
- 8) построить FEO – диаграмму;
- 9) построить IDEF3 – диаграмму;
- 10) произведите стоимостный анализ разработанной модели, сгенерируйте отчет.

Вариант 1.

Разработать модель информационной системы для автоматизации рабочего места библиотекаря.

Вариант 2.

Разработать модель информационной системы для автоматизации рабочего места секретаря школы.

Вариант 3.

Разработать модель информационной системы для автоматизации рабочего места регистратора в поликлинике.

Вариант 4.

Разработать модель информационной системы для автоматизации рабочего места страхового агента.

Вариант 5.

Разработать модель информационной системы для автоматизации рабочего места администратора гостиницы.

Вариант 6.

Разработать модель информационной системы для автоматизации рабочего места кладовщика.

Вариант 7.

Разработать модель информационной системы для автоматизации рабочего места экономиста.

Вариант 8.

Разработать модель информационной системы для автоматизации рабочего места бухгалтера.

Вариант 9.

Разработать модель информационной системы для автоматизации рабочего места инженера.

Вариант 10.

Разработать модель информационной системы для автоматизации рабочего места кассира.

После того как преподаватель определит ваш вариант работы, начинайте его выполнение. Это задание вы будете делать на протяжении всех лабораторных работ, постепенно изучая программную среду BPwin и отрабатывая навыки работы в ней.

Приступая к выполнению контрольного задания:

1. Сформулируйте целевые функции работника предприятия.
2. Определите подразделения предприятия, нормативные документы.
3. Определить входную информацию (данные или материальные ресурсы), которая преобразуется в процессе для получения результата.
4. Определите выходную информацию - готовый результат.
5. Укажите механизмы, которые выполняют процесс.
6. Создайте новый файл в программе **BPwin**.
7. Постройте контекстную диаграмму, произведите ее форматирование.
8. Сохраните новый файл в своей папке с именем **Kr1.bp1**.

 После того, как эта работа будет проделана, пригласите преподавателя и продемонстрируйте ему результат.

Лабораторная работа № 2. Декомпозиция контекстной диаграммы

С помощью этой лабораторной работы Вы сможете:

- научиться производить декомпозицию контекстной диаграммы;

- освоить правила построения дуг и тоннелирования стрелок.

Теоретические сведения

Декомпозиция

 После создания контекстной диаграммы, которая представляет собой описание контекста моделируемой системы, проводится **функциональная декомпозиция** – система разбивается на подсистемы и каждая подсистема описывается в том же синтаксисе, что и система в целом. Затем каждая подсистема разбивается на более мелкие и так до достижения нужного уровня подробности. В результате такого разбиения, каждый фрагмент системы изображается на отдельной диаграмме декомпозиции. *Диаграмма декомпозиции предназначена для детализации работы.*

При декомпозиции процесса все стрелки, входящие или исходящие из него, должны быть перенесены на диаграмму нижнего уровня и использованы при ее построении. При этом запрещены всякие новые стрелки, выходящие за пределы новой диаграммы, кроме специальных, так называемых "тоннелированных" стрелок.

Создание диаграммы А0. Диаграмма верхнего уровня создается путем декомпозиции основной функции контекстной диаграммы. На диаграмме декомпозиции функции нумеруются автоматически слева направо. Номер функции показывается в правом нижнем углу. В левом верхнем исчезает небольшая диагональная черта, которая показывает, что данная функция была декомпозирована.

Практическое задание



Задание 10. Детализация процесса «Изготовление мебели».

Откройте файл **Lab1.bp1**, сохраненный на предыдущем уроке.

Следующим шагом является детализация контекстного процесса с помощью диаграммы верхнего уровня. Эта диаграмма содержит в себе четыре процесса:

- 1) Процесс 1.1 – ПЕРЕРАБОТКА СЫРЬЯ.
- 2) Процесс 1.2 – ИЗГОТОВЛЕНИЕ ДЕТАЛЕЙ.

- 3) Процесс 1.3 – СБОРКА ИЗДЕЛИЯ.
- 4) Процесс 1.4 – КОНТРОЛЬ КАЧЕСТВА.

Произведите детализацию процесса «Изготовление мебели», задав нужное количество новых блоков. Для этого:

1. Щелкните по блоку «Изготовление мебели» и выберите инструмент .
2. В диалоговом окне введите число, на которое будет произведена декомпозиция – 4.
3. Укажите тип диаграммы **IDEF0** (рис. 33) и нажмите ОК.

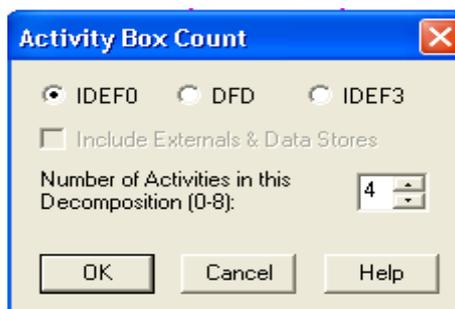


Рисунок 33 - Диалоговое окно декомпозиции блока

4. Укажите названия новых блоков («Переработка сырья», «Изготовление деталей», «Сборка изделия», «Контроль качества»).

При декомпозиции функции, входящие в нее и исходящие из нее дуги, автоматически появляются на диаграмме декомпозиции (миграция дуг), но при этом не касаются блоков. Такие стрелки называются *несвязанными* и воспринимаются в VPwin как синтаксическая ошибка (см. рис.34).

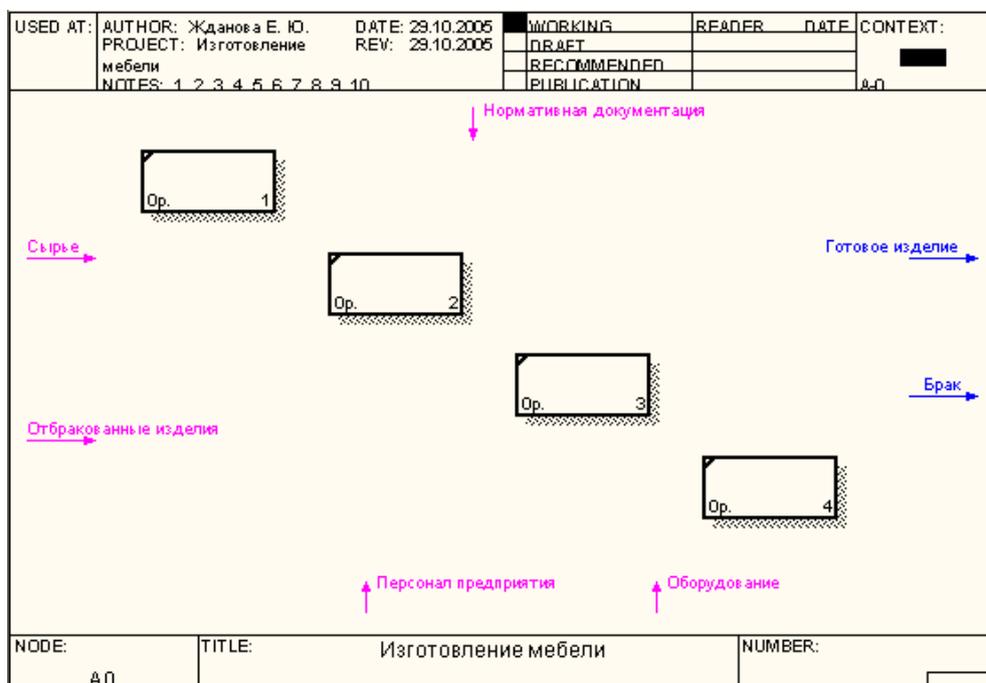


Рисунок 34 - Декомпозиция верхнего уровня

Определим входные и выходные потоки для новых процессов.

Процесс 1.1. ПЕРЕРАБОТКА СЫРЬЯ:

- 1) Вход – СЫРЬЁ.
- 2) Вход – ОТБРАКОВАННЫЕ ИЗДЕЛИЯ.
- 3) Выход – ЗАГОТОВКИ.

Произведем процесс связывания мигрирующих дуг:

5. Выберите инструмент  рисования дуг.
6. Щелкните мышью по наконечнику входного потока СЫРЬЁ.
7. Щелкните по входной стороне блока ПЕРЕРАБОТКА СЫРЬЯ.

Вход – ОТБРАКОВАННЫЕ ИЗДЕЛИЯ построим немного позже.

Для построения выходного потока ЗАГОТОВКИ выполните действия:

8. Выберите инструмент  рисования дуг.
9. Щелкните левой кнопкой мышки по выходной стороне блока ПЕРЕРАБОТКА СЫРЬЯ.

10. Затем щелкните по входной стороне блока ИЗГОТОВЛЕНИЕ ДЕТАЛЕЙ.

11. Выберите инструмент  текст, в контекстном меню – команду **Name**, укажите название дуги ЗАГОТОВКИ.

12. Проверьте себя (рис. 35).

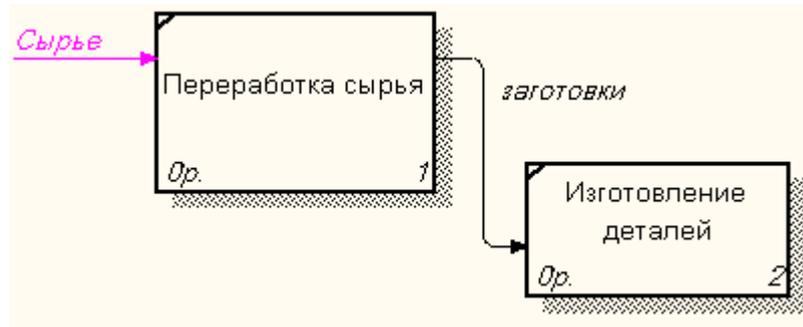


Рисунок 35 - Фрагмент диаграммы



Задание 11. Детализация процесса «Изготовление мебели».

1. Самостоятельно выполните детализацию процессов:

Процесс 1.2. ИЗГОТОВЛЕНИЕ ДЕТАЛЕЙ:

- 1) Вход – ЗАГОТОВКИ.
- 2) Выход – ГОТОВЫЕ ДЕТАЛИ.

Процесс 1.3. СБОРКА ИЗДЕЛИЯ:

- 1) Вход – ГОТОВЫЕ ДЕТАЛИ.
- 2) Выход – СОБРАННОЕ ИЗДЕЛИЕ.

Процесс 1.4. КОНТРОЛЬ КАЧЕСТВА:

- 1) Вход – СОБРАННОЕ ИЗДЕЛИЕ.
- 2) Выход – ГОТОВОЕ ИЗДЕЛИЕ.
- 3) Выход – БРАК.
- 4) Выход – ПРОИЗВОДСТВЕННЫЕ ОТХОДЫ

 После того, как эта работа будет проделана, пригласите преподавателя и продемонстрируйте ему результат.



Задание 12. Смена направления дуги.

На Выходе БРАК не выходит за границу модели, а возвращается в процесс ПЕРЕРАБОТКА СЫРЬЯ:

1. Удалите дуги ОТБРАКОВАННЫЕ ИЗДЕЛИЯ и БРАК.
2. Выберите инструмент  рисование дуг.
3. Щелкните левой кнопкой мыши на Выходе блока КОНТРОЛЬ КАЧЕСТВА.
4. Щелкните левой кнопкой на Входе блока ПЕРЕРАБОТКА СЫРЬЯ.

5. Назовите новую дугу – БРАК (рис. 36).

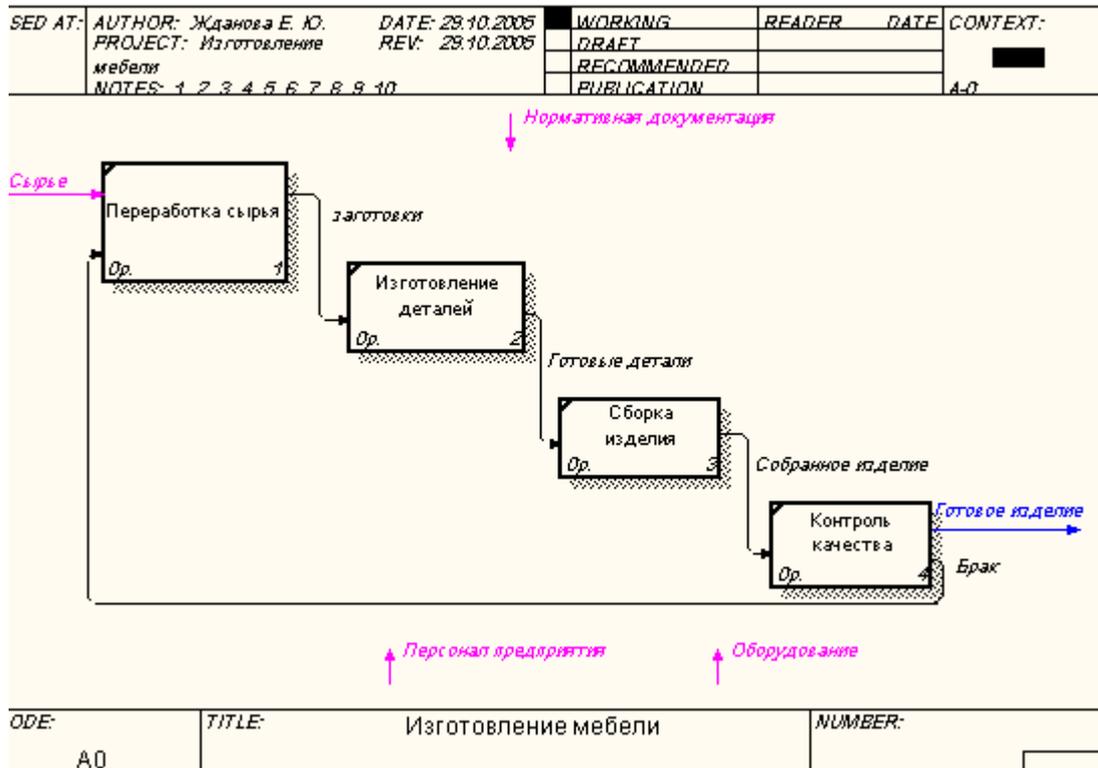


Рисунок 36 - Процесс декомпозиции



Задание 13. Построение ответвлений дуг.

Переработка сырья, изготовление деталей, сборка изделия, контроль качества осуществляются согласно Нормативным документам, поэтому у управляющей стрелки НОРМАТИВНАЯ ДОКУМЕНТАЦИЯ появятся ответвления: НОРМЫ ПЕРЕРАБОТКИ СЫРЬЯ, ЧЕРТЕЖ ДЕТАЛИ, СБОРОЧНЫЙ ЧЕРТЕЖ, СТАНДАРТ КАЧЕСТВА.

1. Выберите инструмент рисование дуг.
2. Щелкните мышью по наконечнику входного потока НОРМАТИВНАЯ ДОКУМЕНТАЦИЯ.
3. Щелкните по входной стороне блока ПЕРЕРАБОТКА СЫРЬЯ.
4. Самостоятельно выполните ответвления дуги НОРМАТИВНАЯ ДОКУМЕНТАЦИЯ на блоки ИЗГОТОВЛЕНИЕ ДЕТАЛЕЙ, СБОРКА ИЗДЕЛИЯ, КОНТРОЛЬ КАЧЕСТВА.
5. Проверьте себя (рис. 37).

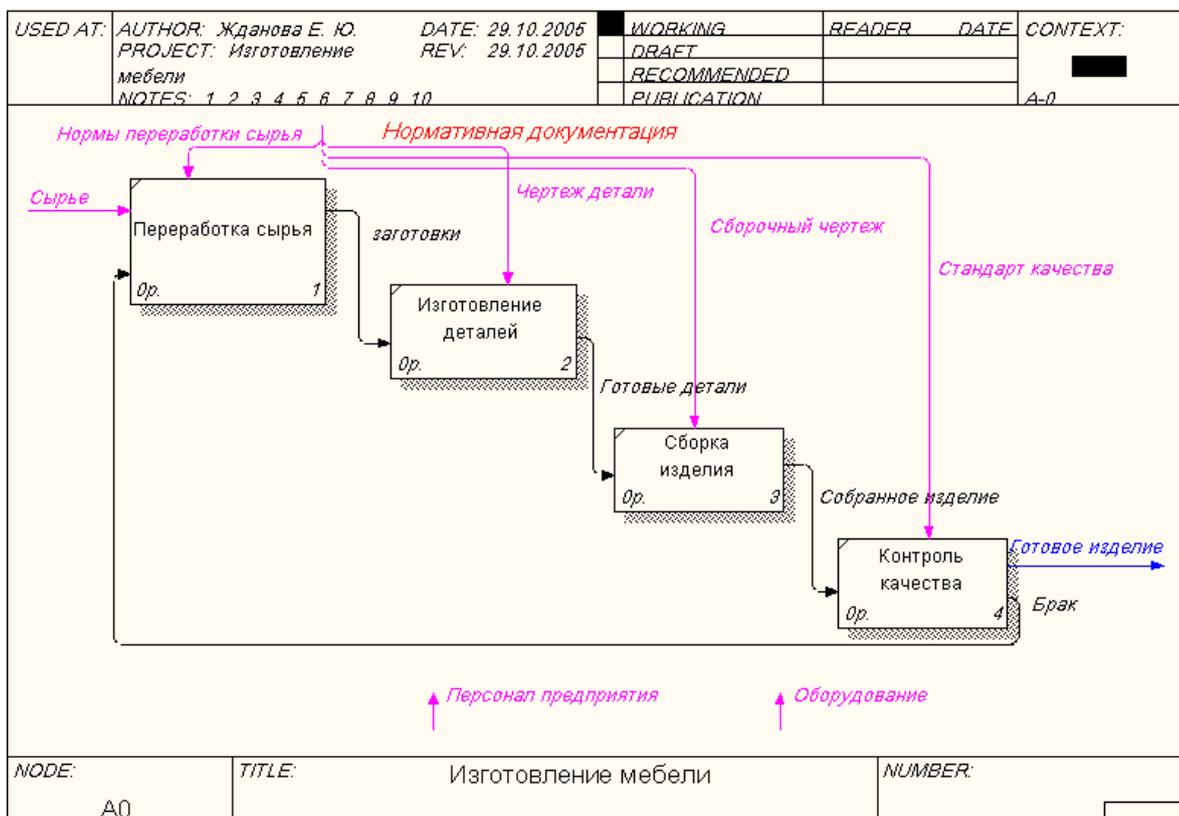


Рисунок 37 - Ответвления дуги НОРМАТИВНАЯ ДОКУМЕНТАЦИЯ



Задание 14. Построение дуг Персонал предприятия, Оборудование.

Дуги ПЕРСОНАЛ ПРЕДПРИЯТИЯ и ОБОРУДОВАНИЕ для всех процессов будут одинаковые.

1. Самостоятельно соедините каждую дугу с каждым блоком, укажите ее имя.
2. Проверьте себя (рис. 38).

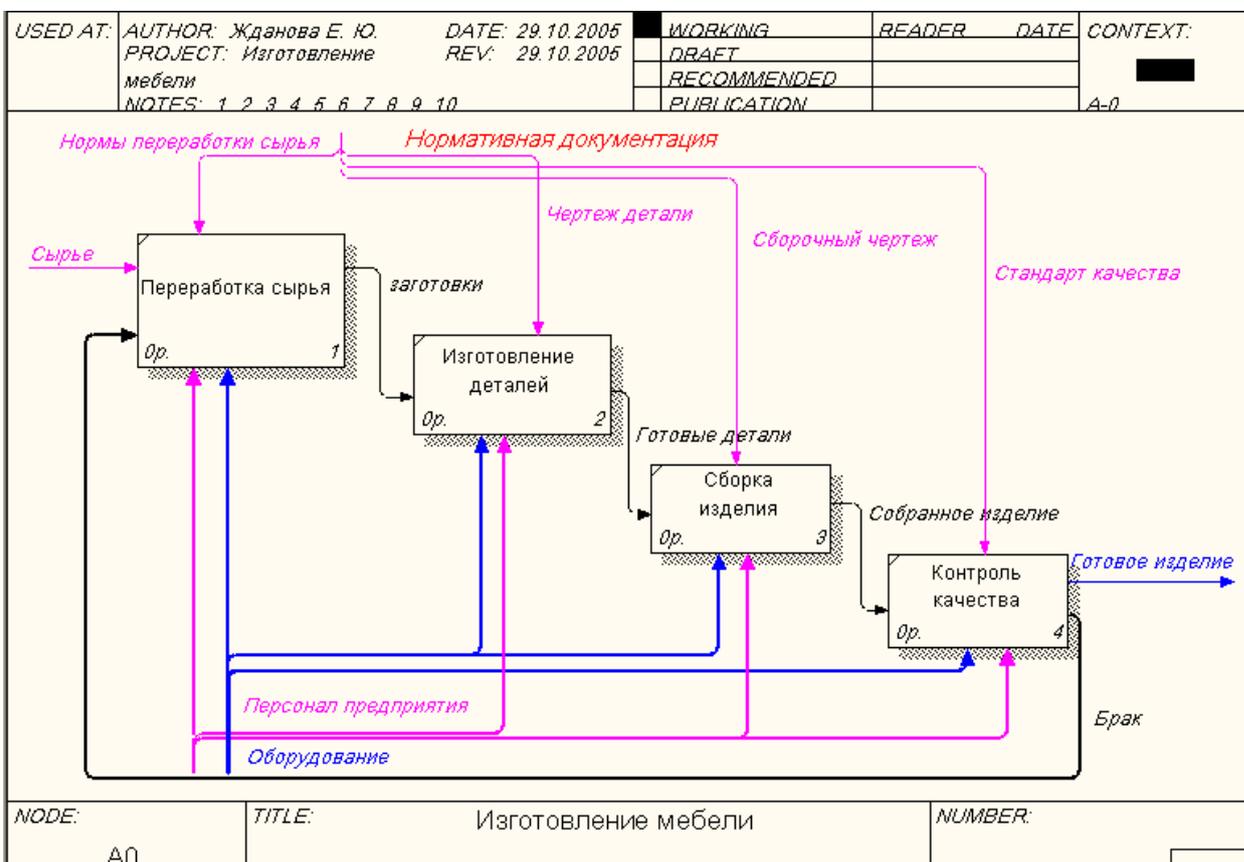


Рисунок 38 - Построение дуг ПЕРСОНАЛ ПРЕДПРИЯТИЯ и
ОБОРУДОВАНИЕ

После того, как эта работа будет проделана, пригласите преподавателя и продемонстрируйте ему результат.



Задание 15. «Тоннелирование» стрелок.

1. В Процессе 1.2. ИЗГОТОВЛЕНИЕ ДЕТАЛЕЙ постройте новую граничную дугу, которой обозначьте Выход – ПРОИЗВОДСТВЕННЫЕ ОТХОДЫ.

Вновь внесенные граничные дуги на диаграмме декомпозиции нижнего уровня изображаются в квадратных скобках и автоматически не появляются на диаграмме верхнего уровня.

Для их «перетаскивания» наверх нужно:

2. Выбрать инструмент редактирования.
3. Кликнуть правой кнопкой мыши по квадратным скобкам.
4. Выбрать в контекстном меню пункт **Arrow Tunnel**.
5. В появившемся диалоге **Border Arrow Editor** (рис. 39) щелкнуть по кнопке **Resolve it to border arrow** для миграции стрелки на диаграмму

верхнего уровня или по кнопке **Change it to resolved rounded tunnel** для «тоннелирования» дуги.



Рисунок 39 - Диалог Border Arrow Editor

Тоннельная дуга изображается с круглыми скобками на конце и не попадет на другую диаграмму (рис.40). Такое тоннелирование может быть применено для изображения малозначимых стрелок.

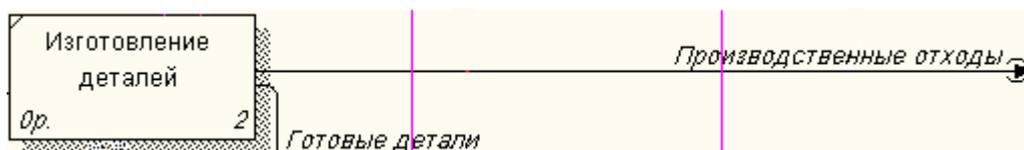


Рисунок 40 - Граничная дуга

6. Отправьте созданную дугу "Производственные отходы" в тоннель.



Задание 16. Создание обратной связи по управлению.

Качество изделия может быть повышено путем непосредственного регулирования процессами изготовления деталей и сборки мебели в зависимости от результата (выхода) работы КОНТРОЛЬ КАЧЕСТВА.

Обратная связь по управлению свидетельствует об эффективности бизнес-процесса и создается следующим образом:

1. Выберите инструмент  рисование дуг.
2. Щелкните мышью по выходу КОНТРОЛЬ КАЧЕСТВА.

3. Щелкните по управлению блоков ИЗГОТОВЛЕНИЕ ДЕТАЛЕЙ, СБОРКА ИЗДЕЛИЯ.

4. Выберите инструмент **T** текст.

5. Назовите обратную связь РЕКОМЕНДАЦИИ.

После выполнения работы у вас должна получиться следующая диаграмма (рис. 41).

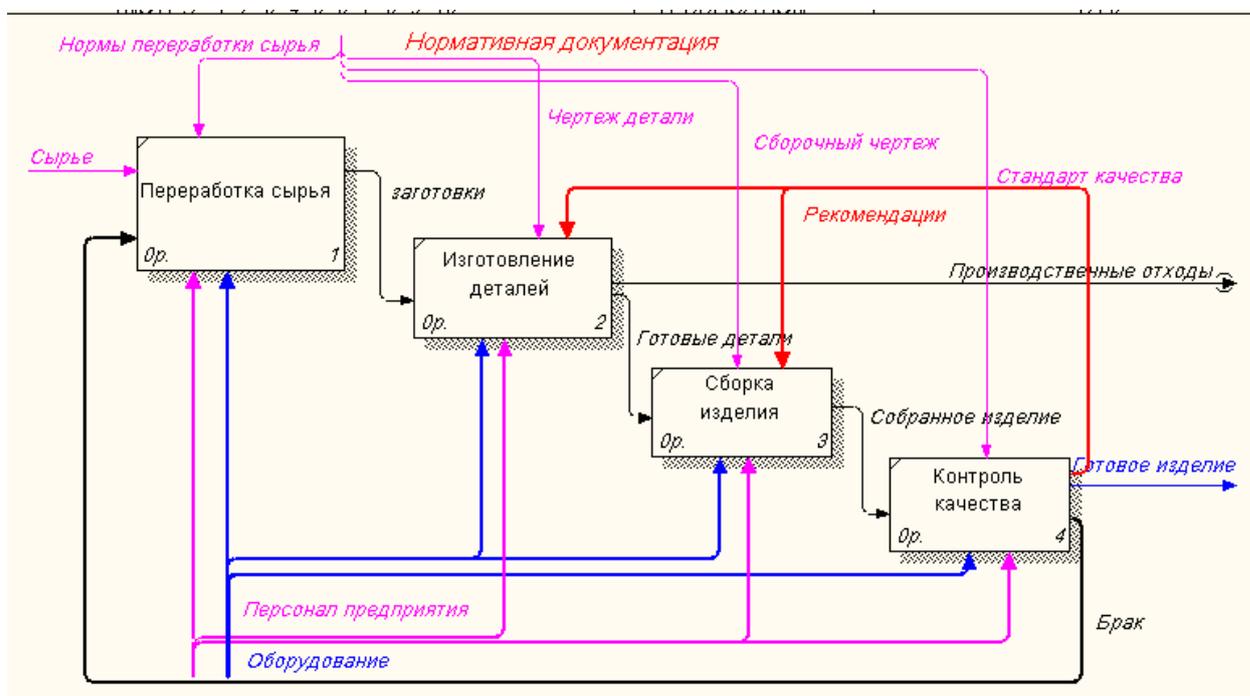


Рисунок 41 - Диаграмма декомпозиции блока ИЗГОТОВЛЕНИЕ МЕБЕЛИ



Задание 17. Сохранение полученной диаграммы.

1. В меню **File** выберите **Save as**.
2. Укажите путь к своей папке и имя файла **Lab2.bp1**.
3. Нажмите ОК.

После того, как эта работа будет проделана, пригласите преподавателя и продемонстрируйте ему результат.

Контрольные вопросы

1. Как создается диаграмма верхнего уровня?
2. Как на диаграмме отображается декомпозиция?
3. Каким инструментом задается количество блоков для декомпозиции?

4. Какие стрелки WinVisio воспринимает как синтаксическую ошибку?
5. Для чего создается обратная связь по управлению?
6. Для чего служит туннелирование дуг?

 **Тестовое задание**

1. В каком порядке надо располагать блоки на диаграмме детализации:

- a) в произвольном;
- b) обязательно справа налево;
- c) в соответствии с уровнем сложности последующей детализации;
- d) в соответствии с их доминированием;
- e) все ответы правильные.

2. Для чего предназначена диаграмма детализации процесса:

- a) для обозначения туннелированных дуг;
- b) для описания контекста моделируемой системы;
- c) для детализации работы;
- d) все ответы правильные.

3. Как создается диаграмма верхнего уровня:

- a) путем декомпозиции одной из функций контекстной диаграммы;
- b) путем декомпозиции основной функции контекстной диаграммы;
- c) путем декомпозиции диаграмм нижнего уровня;
- d) путем декомпозиции одной из функций на диаграмме верхнего уровня;
- e) правильного ответа нет.

4. Какой инструмент необходимо выбрать, чтобы с его помощью произвести декомпозицию блока?

- a)  ;
- b)  ;
- c)  ;
- d)  ;
- e)  .

5. Каким инструментом производится процесс связывания мигрирующих дуг?

- a)  ;
- b)  ;
- c)  ;
- d)  ;
- e)  .

- б. Мигрирующая дуга – это дуга, которая...**
- а) создает обратную связь по управлению;
 - б) переходит из родительской диаграммы и не касается блока;
 - с) обязательно тоннелируется;
 - д) создает обратную связь по входу;
 - е) правильного ответа нет.

 *После того, как эта работа будет проделана, пригласите преподавателя и продемонстрируйте ему результат.*

Контрольное задание

Продолжите выполнение контрольного задания.

1. Откройте файл **Kr1.bp1**.
2. Произведите детализацию бизнес-процесса.
3. Сохраните файл в своей папке с именем **Kr2.bp1**.

 *После того, как эта работа будет проделана, пригласите преподавателя и продемонстрируйте ему результат.*

Лабораторная работа № 3. Построение функциональной модели.

Описание модели

С помощью этой лабораторной работы Вы сможете:

- научиться детализировать процессы;
- освоить правила описания свойств модели;
- научиться составлять отчет о свойствах модели.

Теоретические сведения

 Последним шагом построения модели является **функциональная декомпозиция**. Построенная диаграмма верхнего уровня также имеет множество процессов, которые в свою очередь могут быть детализированы в диаграммы нижнего уровня. Таким образом строится иерархия IDEF0 с контекстной диаграммой в вершине иерархии.

Этот процесс декомпозиции продолжается до достижения нужного уровня подробности. При таком построении иерархии IDEF0 каждый процесс более низкого уровня необходимо соотнести с процессом верхнего уровня.

Обычно для этой цели все работы модели нумеруются. Номер состоит из префикса и числа. Может быть использован префикс любой длины, но обычно используют префикс А.

Контекстная 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**.

Имеются незначительные варианты нумерации, которые можно настроить во вкладке **Numbering** (рис. 42) диалога **Model Properties** (меню **Model – Model Properties**).

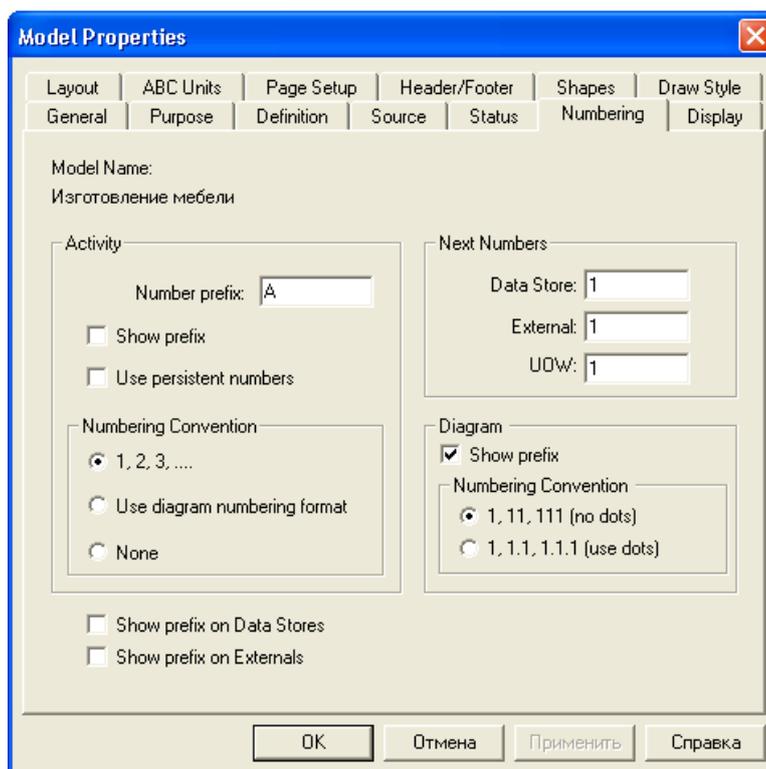


Рисунок 42 - Диалоговое окно настройки нумерации работ в диаграмме

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, A1, A2, A21, A213 и т.д.).

ВРwin 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. ВРwin позволяет иметь в модели только одну диаграмму декомпозиции в данном узле. Прежние 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 FEO, 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 - это произвольная строка, но рекомендуется придерживаться стандарта, когда номер состоит из буквенного префикса и порядкового номера, причем в качестве префикса используются инициалы автора диаграммы, а порядковый номер отслеживается автором вручную, например **ЖЕЮ00021** (рис. 43).

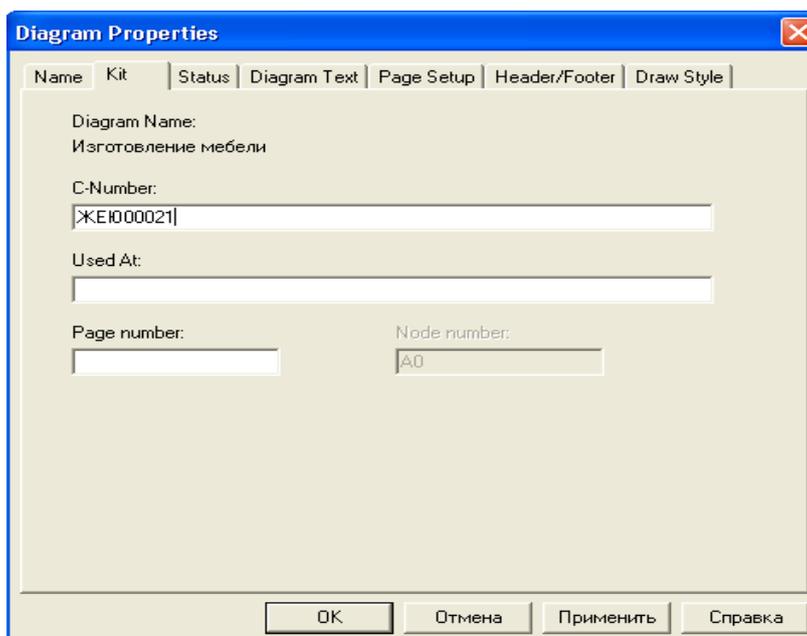


Рисунок 43 - Диалоговое окно присваивания номера данной версии диаграммы

Практическое задание

На предыдущих лабораторных работах вы построили контекстную диаграмму процесса "Изготовление мебели" и провели его детализацию с помощью диаграммы верхнего уровня. Последним шагом построения модели является **функциональная декомпозиция**, т.е. разбиение сложных процессов на более простые. Этот процесс декомпозиции продолжается до достижения нужного уровня подробности.



Задание 18. Детализация процесса «Изготовление деталей».

1. Откройте файл **Lab2.bp1**, сохраненный на предыдущем уроке.
2. Проведите детализацию процесса **1.2. ИЗГОТОВЛЕНИЕ ДЕТАЛЕЙ** с помощью диаграммы нижнего уровня. Данные представлены в таблице 4.

Таблица 4 - Детализирование процесса «Изготовление деталей»

Процесс	Вход	Выход
1.2.1 – Переработка заготовки в деталь	Заготовки	Готовые детали
1.2.2 – Проверка качества деталей	Готовые детали	Готовые детали, брак

Управляющие стрелки и стрелки механизмов, указанные на диаграмме верхнего уровня должны быть и в диаграмме детализации.

3. Выберите инструмент  и щелкните по блоку **ИЗГОТОВЛЕНИЕ ДЕТАЛЕЙ**;
4. В диалоговом окне введите число, на которое будет произведена декомпозиция - 2;
5. Укажите тип диаграммы **IDEF0** (рис. 44) и нажмите ОК.

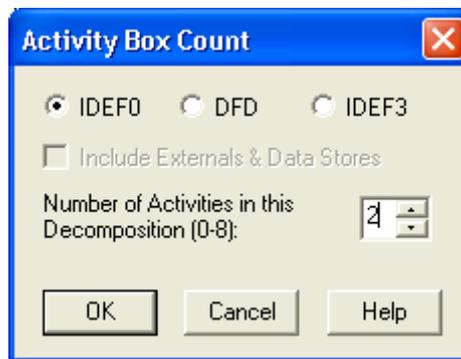


Рисунок 44 - Диалоговое окно декомпозиции блока

Вы получите диаграмму декомпозиции уровня **A2** (рис. 45).

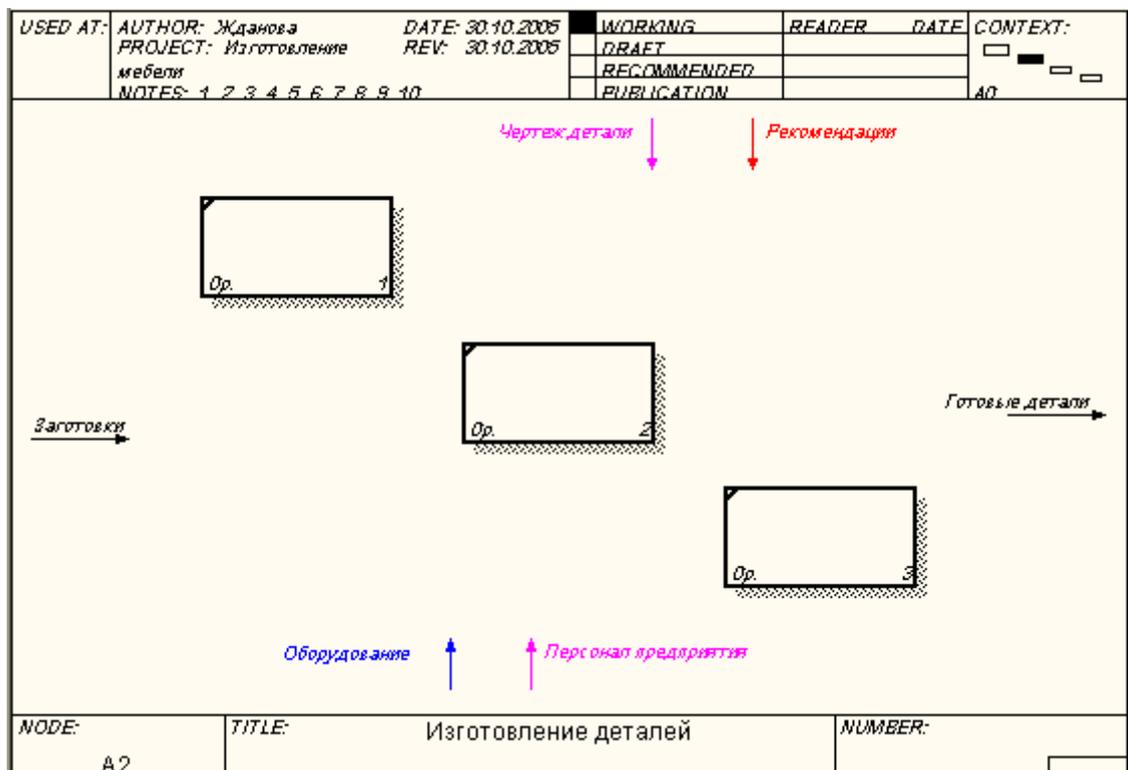


Рисунок 45 - Декомпозиция уровня A2

6. Укажите названия процессов;
7. Соедините дугами обозначенные процессы, используя данные из таблицы 3;
8. Проверьте себя (рис. 46).

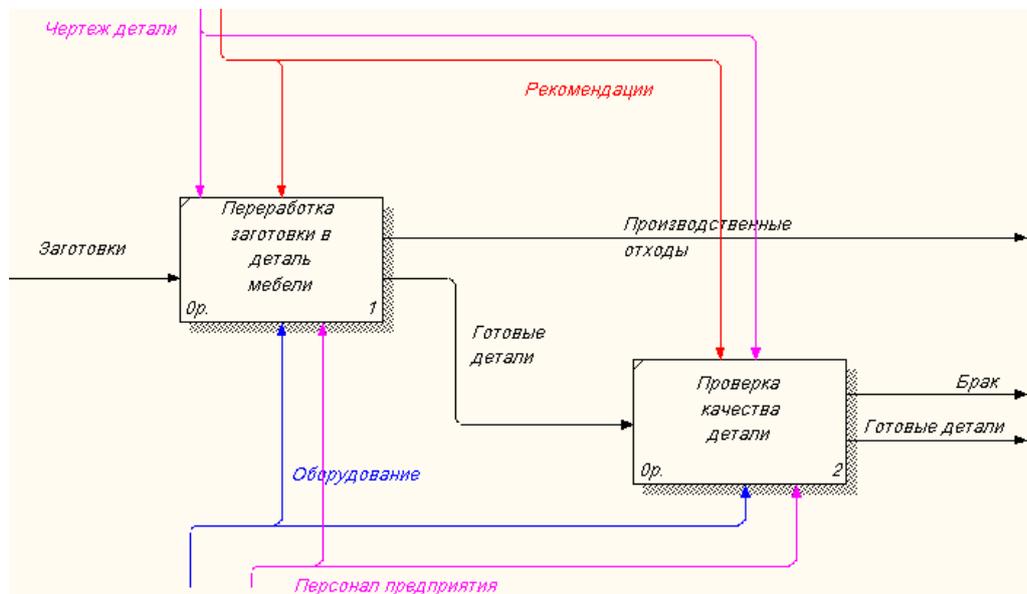


Рисунок 46 - Детализация процесса ИЗГОТОВЛЕНИЕ ДЕТАЛЕЙ



Задание 19. Детализация процесса «Контроль качества».

1. Самостоятельно выполните детализацию процесса КОНТРОЛЬ КАЧЕСТВА.

После выполнения работы у вас должна получиться следующая диаграмма (рис. 47).

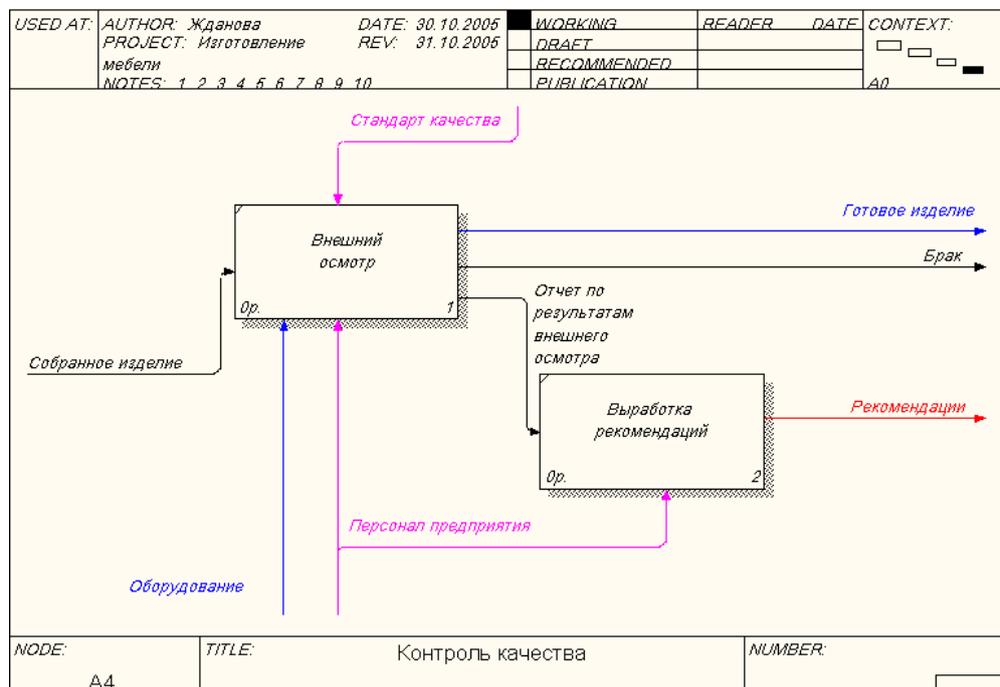


Рисунок 47 - Детализация процесса КОНТРОЛЬ КАЧЕСТВА

После того, как эта работа будет проделана, пригласите преподавателя и продемонстрируйте ему результат.



Задание 20. Описание свойств модели.

IDEF0-модель предполагает наличие четко сформулированной цели, единственного субъекта моделирования и одной точки зрения. Для внесения цели, цели и точки зрения в модели IDEF0 в VPwin следует:

1. Выбрать пункт меню **Model - Model Properties**, вызывающий диалог **Model Properties** (рис. 48);

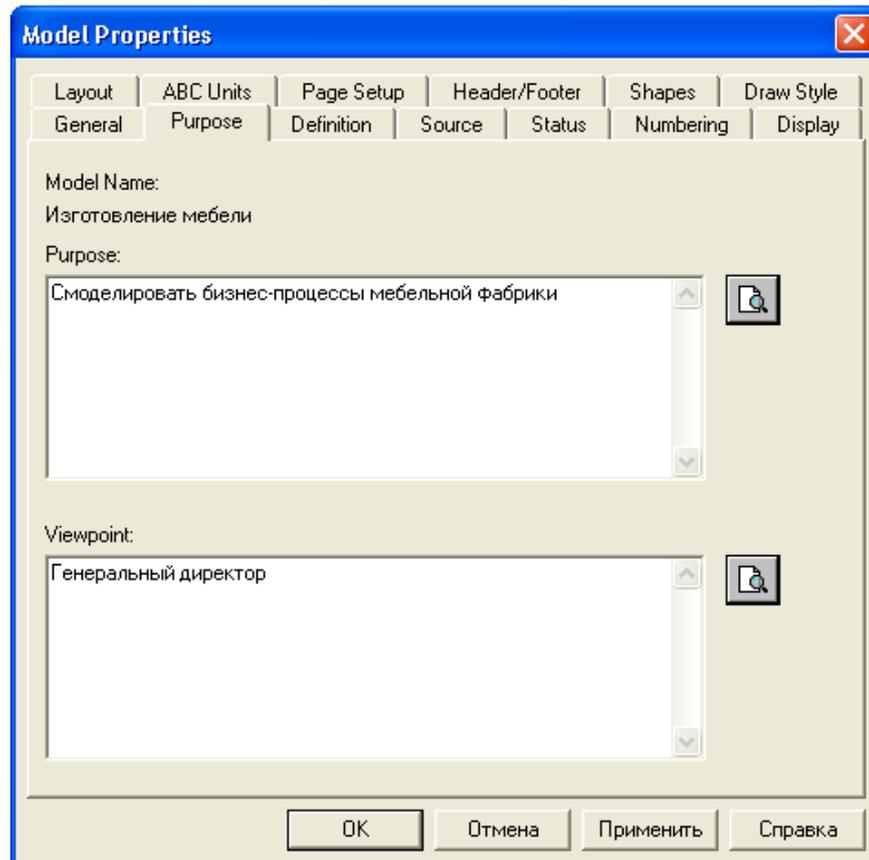


Рисунок 48 - Диалог задания свойств модели

2. Во вкладку **Purpose** внести цель и точку зрения, а во вкладку **Definition** – определение модели;

Цель и точку зрения принято выносить на контекстную диаграмму **A-0** в виде текстового блока. После описания они появятся на контекстной диаграмме в виде текстового блока. Описание производится на уровне контекстной диаграммы.

Для описания цели и точки зрения следует:

3. Перейти на уровень диаграммы **A-0**;
4. Выбрать кнопку текста **T** на палитре инструментов;

5. Щелкнуть мышью в позиции предполагаемого ввода текста;
6. В диалоговом окне набрать нужный текст и установить опцию значимости (обычный текст, цель или точка зрения) (рис. 49).

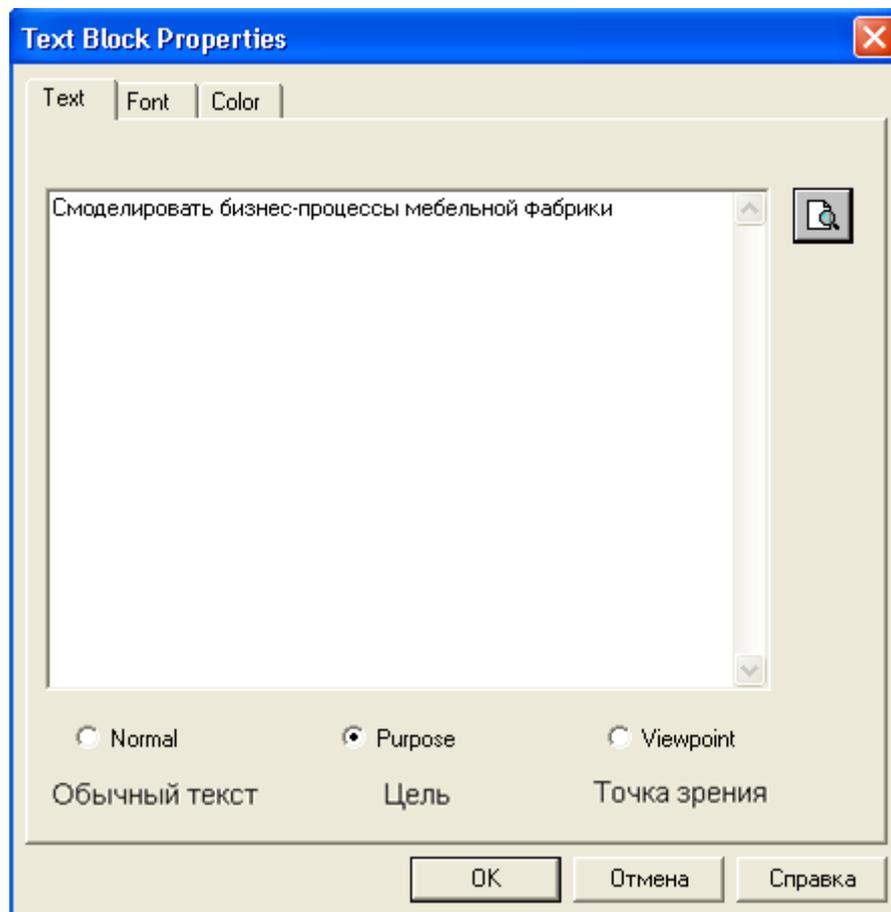


Рисунок 49 - Установление опции Text

7. Во вкладке **Status** того же диалога опишите статус модели (черновой вариант, рабочий, окончательный и т.д.), время создания и последнего редактирования (отслеживается в дальнейшем автоматически по системной дате);

8. Во вкладке **Source** опишите источники информации для построения модели (например, «Опрос экспертов предметной области и анализ документации»);

9. Вкладка **General** служит для внесения имени проекта и модели, имени и инициалов автора и временных рамок модели.



Задание 21. Составление отчета.

Результат описания модели можно получить в отчете **Model Report**.

1. Диалоговое окно настройки отчета по модели вызовите из пункта меню **Tools – Reports - Model Report**.
2. Выберите необходимые поля, при этом автоматически отображается очередность вывода информации в отчете (рис. 50);

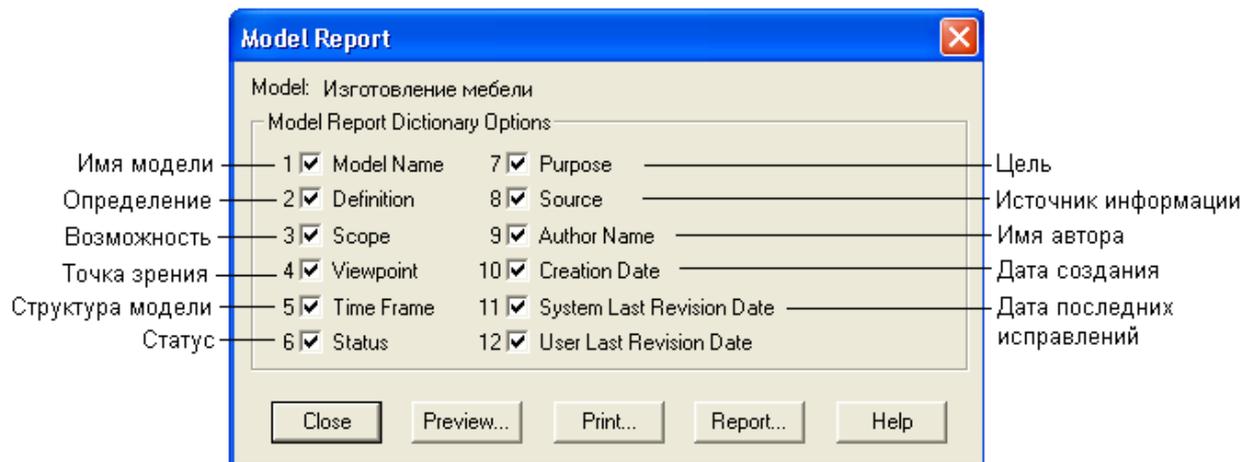


Рисунок 50 - Диалоговое окно выбора информации для отчета

3. Нажмите на кнопку **Preview**, чтобы просмотреть отчет (рис.51).

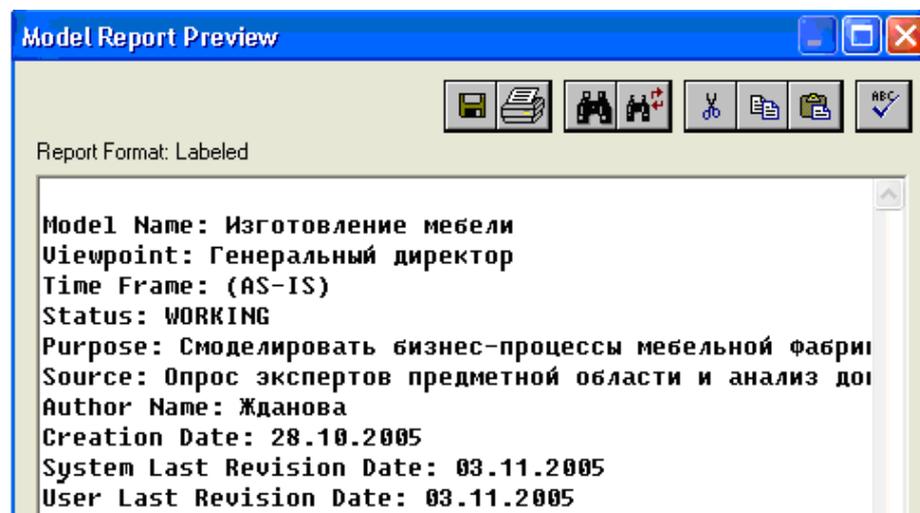


Рисунок 51 - Отчет о модели



Задание 22. Сохранение полученной диаграммы.

1. В меню **File** выберите **Save As**.
2. Укажите путь к своей папке и имя файла **Lab3.bp1**.
3. Нажмите **ОК**.

 После того, как эта работа будет проделана, пригласите преподавателя и продемонстрируйте ему результат.

Контрольные вопросы

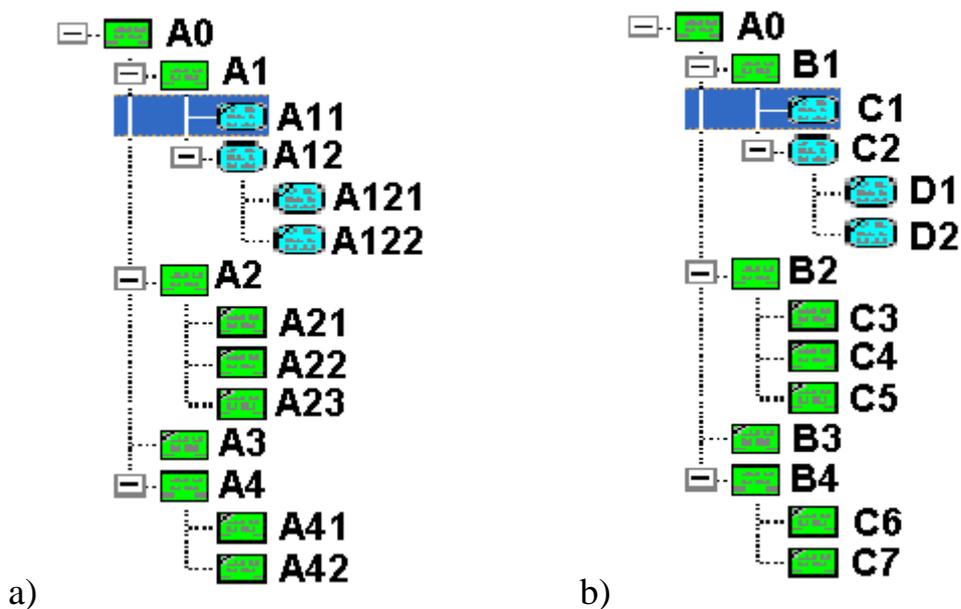
1. Как нумеруются модели в иерархии **IDEF0**?
2. Дайте понятие определению **Дерево узлов**.
3. Какой процесс в разработке модели называют функциональной декомпозицией?
4. Как можно вынести цель и точку зрения проекта на диаграмму?
5. Для чего необходимо составление отчета?

🔑 Тестовое задание

1. **IDFE0** - это:

- a) диаграмма потока данных;
- b) диаграмма бизнес - процесса;
- c) диаграмма сущность - связь;
- d) диаграмма ключей;
- e) все ответы правильные.

2. Укажите номер рисунка, который отражает правильную нумерацию по узлам:



3. Сколько контекстных диаграмм может быть в модели:

- a) одна;
- b) две;
- c) столько, сколько уровней детализации;

- d) неограниченное количество;
- e) все ответы правильные.

4. Дуги управления указывают на:

- a) готовый результат;
- b) промежуточный результат;
- c) ресурсы, которые выполняют процесс;
- d) правила, стандарты;
- e) информацию для получения результата;
- f) все ответы правильные.

5. Какой инструмент позволяет внести цель и точку зрения на контекстную диаграмму?

- a)  ;
- b)  ;
- c)  ;
- d)  ;
- e)  .

 После того, как эта работа будет проделана, пригласите преподавателя и продемонстрируйте ему результат.

Контрольное задание

Продолжите выполнение контрольного задания.

1. Откройте файл **Kr2.bp1**.
2. Произведите декомпозицию 2-х бизнес-процессов.
3. Создайте описание модели.
4. Сохраните файл в своей папке с именем **Kr3.bp1**.

 После того, как эта работа будет проделана, пригласите преподавателя и продемонстрируйте ему результат.

Лабораторная работа № 4. Построение диаграммы дерева узлов и FEO

С помощью этой лабораторной работы Вы сможете:

- освоить принципы построения диаграммы дерева узлов;
- научиться задавать свойства и стиль диаграмме дерева узлов;

- освоить правила построения диаграммы FEO.

Теоретические сведения

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 (рис. 52).

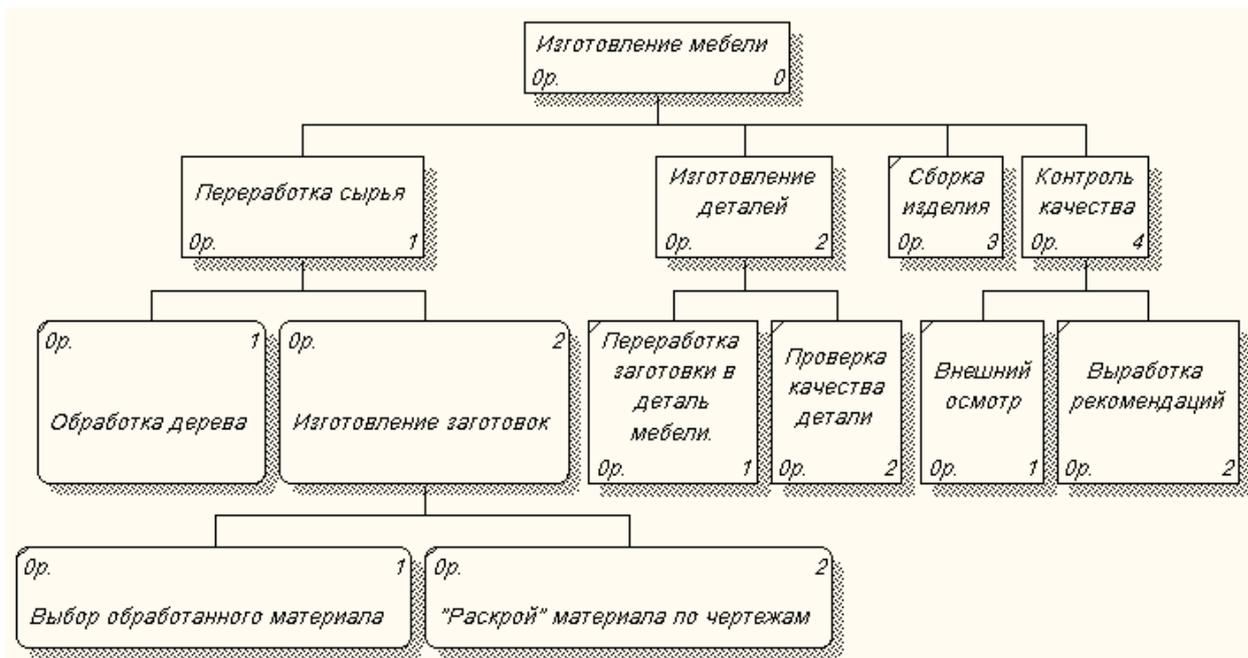


Рисунок 52 - Диаграмма дерева узлов

Процесс создания модели работ является итерационным (повторяющимся, многократно меняющимся), 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** (рис. 53), который позволяет представить иерархию работ и диаграмм в удобном и компактном виде, однако этот инструмент не является составляющей стандарта **IDEF0**.



Рисунок 53 - Навигатор модели Model Explorer

Практическое задание



Задание 23. Создание диаграммы дерева узлов.

1. Откройте файл **Lab3.bp1**, сохраненный на предыдущем уроке.

Для создания диаграммы дерева узлов следует:

2. Выбрать в меню пункт **Diagram - Add Node Tree**.

Появится диалог создания диаграммы дерева узлов **Node Tree Wizard** (рис. 54).



Рисунок 54 - Диалог создания диаграммы дерева узлов Node Tree Wizard

3. В первом диалоге эксперта введите имя диаграммы дерева узлов, узел верхнего уровня и глубину дерева – **Number of Levels** (по умолчанию 3).

Дерево узлов не обязательно в качестве верхнего уровня должно иметь контекстную работу и может иметь произвольную глубину. В одной модели можно создавать множество диаграмм деревьев узлов.

Имя дерева узлов по умолчанию совпадает с именем работы верхнего уровня, а номер диаграммы автоматически генерируется как номер узла верхнего уровня плюс литера "N", например, A0N.

Второй диалог эксперта **Node Tree Wizard** (рис. 55) позволяет задать свойства диаграммы дерева узлов.



Рисунок 55 - Диалог эксперта Node Tree Wizard

По умолчанию нижний уровень декомпозиции показывается в виде списка, остальные работы - в виде прямоугольников (рис. 56).

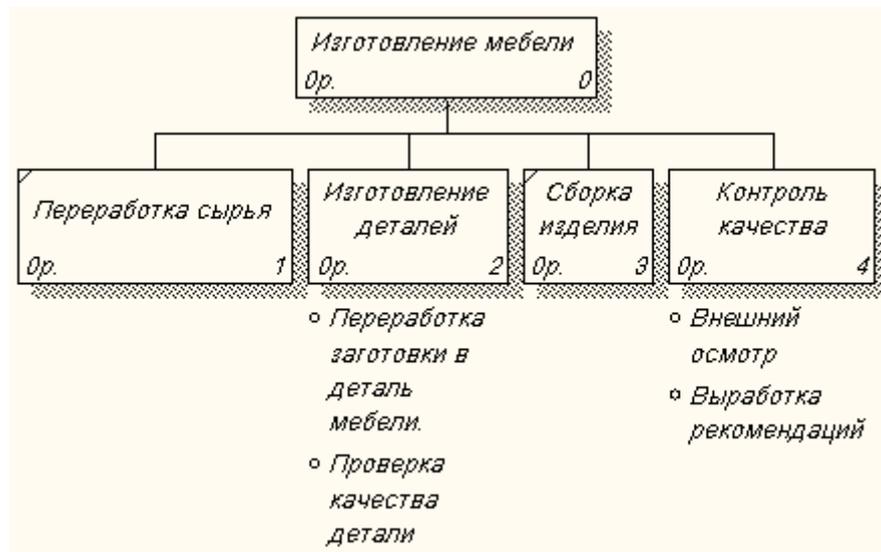


Рисунок 56 - Дерево узлов с ортогональными линиями

Для отображения всего дерева в виде прямоугольников следует убрать опцию **Bullet Last Level**. Группа **Connection Style** позволяет выбрать стиль соединительных линий - диагональные (по умолчанию) или ортогональные.

4. Щелкните правой кнопкой мыши по свободному месту, не занятому объектами, выберите меню **Node tree Diagram Properties** (рис. 57).

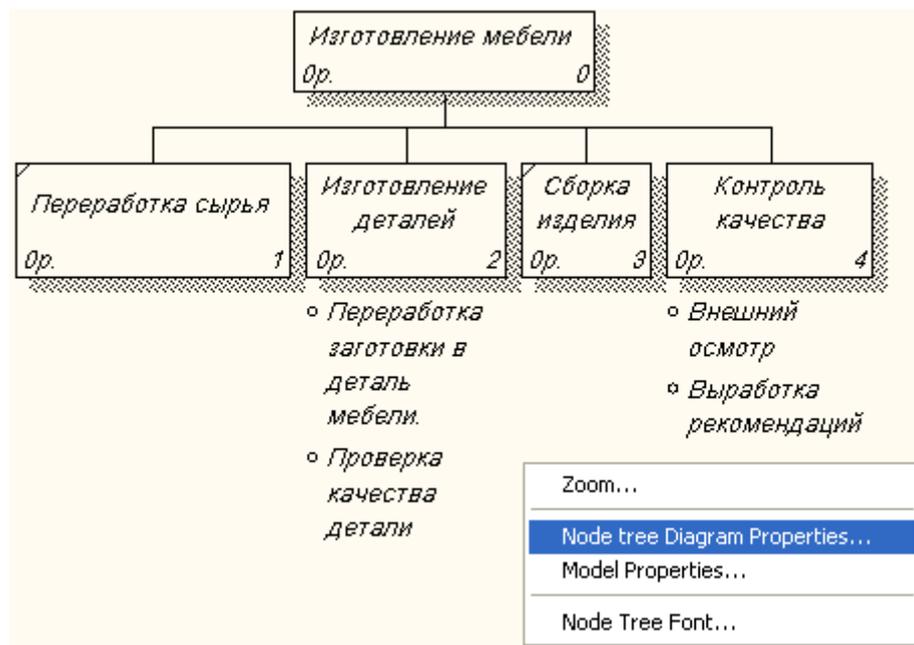


Рисунок 57 - Выбор меню Node tree Diagram Properties

5. Во вкладке **Style** диалога **Node Tree Properties** отключите опцию **Bullet Last Level** (рис. 58).

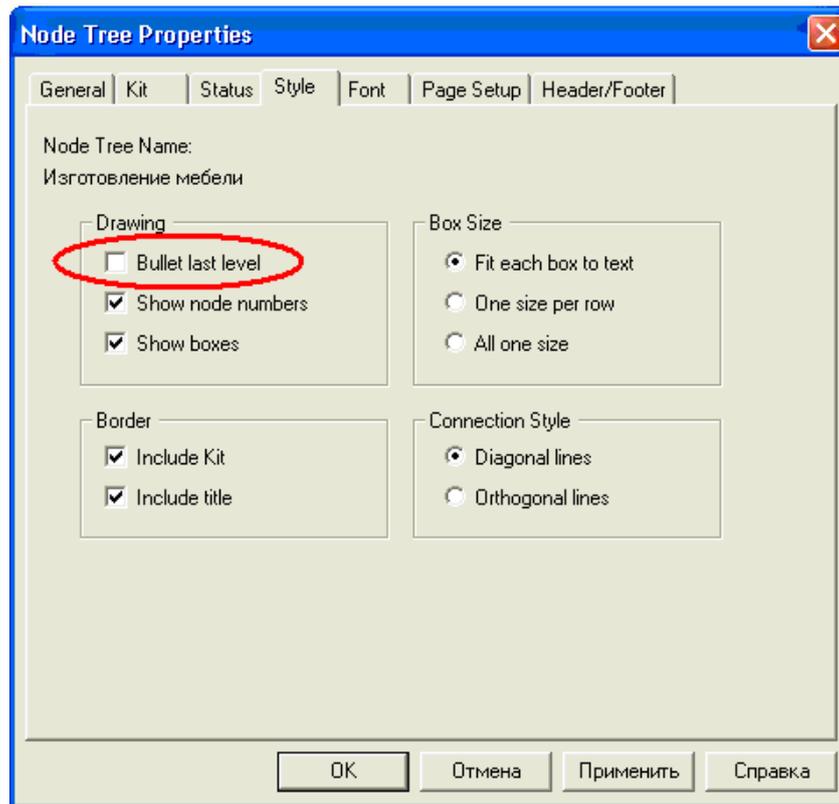


Рисунок 58 - Диалоговое окно Node Tree Properties

6. Щелкните по кнопке **ОК**.
7. Проверьте полученный результат (рис. 59).

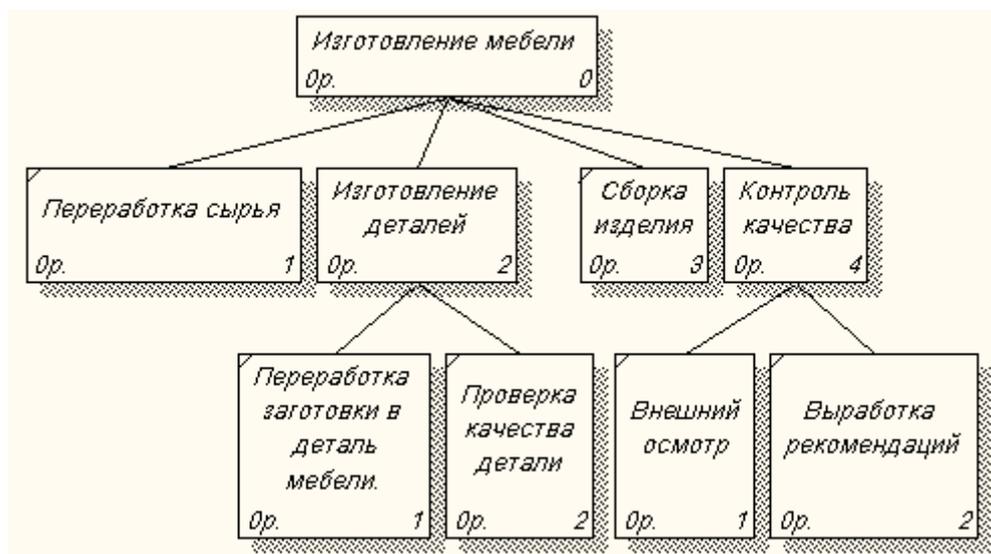


Рисунок 59 - Дерево узлов

8. Самостоятельно создайте диаграмму **Дерево узлов** с ортогональными линиями к работе «Изготовление деталей».

9. Проверьте себя (рис. 60).

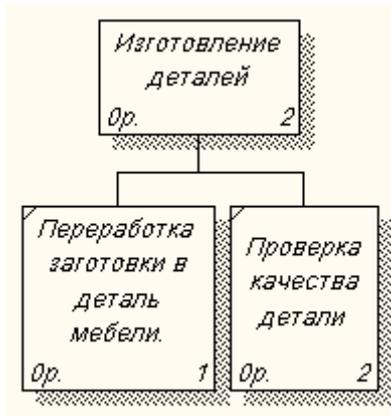


Рисунок 60 - Дерево узлов к работе «Изготовление деталей»

 После того, как эта работа будет проделана, пригласите преподавателя и продемонстрируйте ему результат.



Задание 24. Создание диаграммы FEO.

Диаграммы "только для экспозиции" (FEO) часто используются в модели для иллюстрации других точек зрения, для отображения отдельных деталей, которые не поддерживаются явно синтаксисом IDEF0, они по сути являются просто картинками - копиями стандартных диаграмм и не включаются в анализ синтаксиса.

Создайте диаграмму FEO.

1. Выберите пункт меню **Diagram - Add FEO Diagram**.
2. Укажите имя диаграммы FEO и тип родительской диаграммы (рис. 61).
3. Нажмите на кнопку **ОК**.

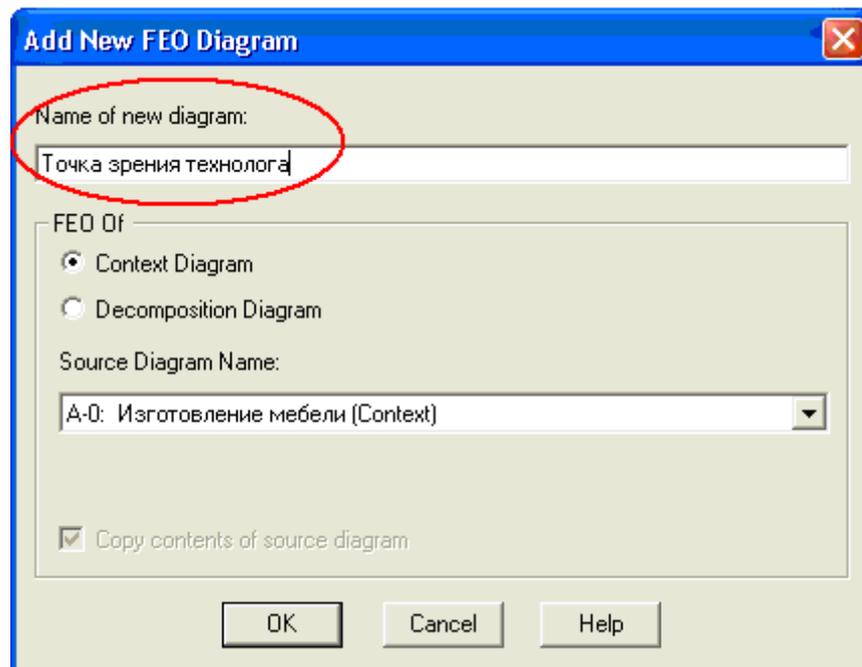


Рисунок 61 - Диалоговое окно Add New FEO Diagram

У вас получится «картинка» контекстной диаграммы «Изготовление мебели» (рис. 62).



Рисунок 62 - Иллюстрация к контекстной диаграмме «Изготовление мебели»

4. Самостоятельно создайте диаграмму FEO для диаграммы декомпозиции «Контроль качества».

5. Проверьте себя (рис. 63).

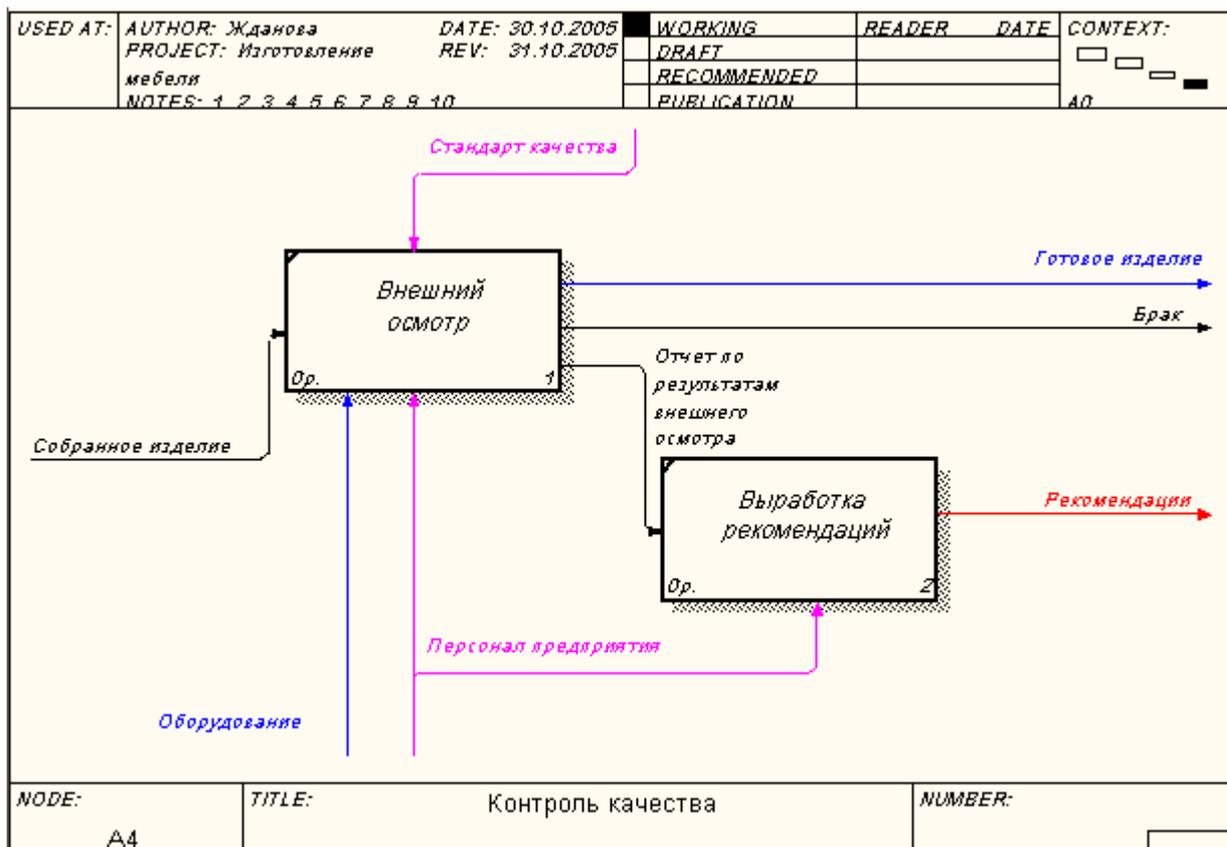


Рисунок 63 - Иллюстрация к диаграмме декомпозиции «Контроль качества»



Задание 25. Сохранение полученной диаграммы.

1. В меню **File** выберите **Save As**.
2. Укажите путь к своей папке и имя файла **Lab4.bp1**.
3. Нажмите ОК.

👤 После того, как эта работа будет проделана, пригласите преподавателя и продемонстрируйте ему результат.

Контрольные вопросы

1. Для чего строится диаграмма дерева узлов?
2. Какие свойства и стиль можно задать диаграмме дерева узлов?
3. Сколько диаграмм дерева узлов можно построить к одной модели?

4. Для чего используются диаграммы FEO?
5. Чем отличаются друг от друга диаграммы дерева узлов и FEO?

💡 **Тестовое задание**

1. Какая диаграмма позволяет представить иерархию работ в удобном и компактном виде?

- a) контекстная;
- b) дерево узлов;
- c) функциональная;
- d) FEO;
- e) все ответы правильные.

2. Сколько можно построить диаграмм дерева узлов к одной модели?

- a) одну;
- b) две;
- c) несколько;
- d) только к контекстной диаграмме;
- e) правильные ответы a), d).

3. Какие опции необходимо указать в диалоговом окне Node Tree Wizard, чтобы построить диаграмму дерева узлов:

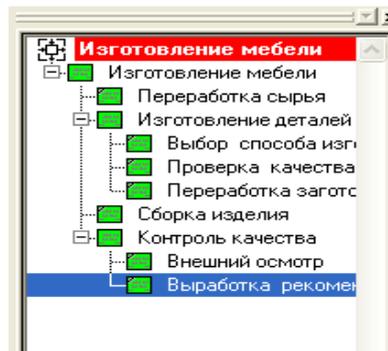
- a) имя диаграммы;
- b) глубину дерева;
- c) узел верхнего уровня;
- d) цвет диаграммы;
- e) все ответы правильные.

4. Для отображения всего дерева в виде прямоугольников следует убрать опцию:

- a) Include title;
- b) Show boxes;
- c) Bullet Last Level;
- d) Show node numbers;
- e) Orthogonal lines.

5. Как называется элемент окна программы VPwin, изображенный на рисунке?

- a) панель инструментов редактирования;
- b) рабочая зона;
- c) навигатор модели;
- d) дерево узлов;
- e) правильного ответа нет.



 После того, как эта работа будет проделана, пригласите преподавателя и продемонстрируйте ему результат.

Контрольное задание

Продолжите выполнение контрольного задания.

1. Откройте файл **Kr3.bp1**.
2. Постройте диаграмму дерева узлов к контекстной диаграмме, указав её имя и глубину дерева.
3. Постройте диаграмму **FEO**.
4. Сохраните файл в своей папке с именем **Kr4.bp1**.

 После того, как эта работа будет проделана, пригласите преподавателя и продемонстрируйте ему результат.

Лабораторная работа № 5. Построение диаграммы потоков данных

С помощью этой лабораторной работы Вы сможете:

- освоить назначение и принципы построения DFD - диаграммы потоков данных;
- научиться устанавливать внутренние ссылки;
- научиться производить детализацию диаграммы DFD.
-

Теоретические сведения

Диаграммы DFD

 **Диаграммы потоков данных** (Data flow diagramming, **DFD**) можно использовать как дополнение к модели **IDEF0** для более *наглядного*

отображения текущих операций документооборота в системах обработки информации.

Диаграммы потоков данных используются для описания документооборота и обработки информации и представляют модельную систему как *сеть связанных между собой работ*.

Диаграммы потоков данных (**DFD**) показывают *внешние источники и приемники данных, потоки данных и хранилища (накопители) данных*, к которым осуществляется доступ.

DFD описывает:

- 1) функции обработки информации (работы);
- 2) документы (стрелки, *arrows*), объекты, сотрудников или отделы, которые участвуют в обработке информации;
- 3) внешние ссылки (*external references*), которые обеспечивают интерфейс с внешними объектами, находящимися за границами моделируемой системы;
- 4) таблицы для хранения документов (хранилища данных, *data store*).

Для изображения диаграмм потоков данных в **ВРwin** используется нотация **Гейна-Сарсона** (рис. 64).

компонента	нотация Гейна-Сарсона
поток данных	ИМЯ →
управляющий процесс	НОМЕР ИМЯ
хранилище данных	№ ИМЯ
внешняя сущность	НОМЕР ИМЯ

Рисунок 64 - Нотация Гейна-Сарсона

Этапы построения модели

1. Выделение внешних объектов, с которыми система должна быть связана.

2. Формирование DFD диаграммы первого уровня.
3. Функциональная декомпозиция каждого подпроцесса с помощью диаграмм нижнего уровня.
4. Составление словаря данных.
5. Построение спецификаций процесса, если его нельзя выразить комбинацией подпроцессов.

При дополнении модели IDEF0 диаграммой DFD, в палитре инструментов на новой диаграмме **DFD** появляются новые кнопки:



- добавить в диаграмму внешнюю ссылку (*External Reference*). Внешняя ссылка является источником или приемником данных извне модели.



- добавить в диаграмму хранилище данных (*Data store*). Хранилище данных позволяет описать данные, которые необходимо сохранить в памяти прежде, чем использовать в работах.

В отличие от IDEF0, где система рассматривается как взаимосвязанные работы, DFD рассматривает систему как совокупность предметов.

Работы в DFD представляют собой функции системы, преобразующие *входы в выходы*. Хотя работы изображаются прямоугольниками со скругленными углами, смысл их совпадает со смыслом работ в IDEF0, они имеют входы и выходы, но не поддерживают управления и механизмы, как IDEF0 (рис. 65).

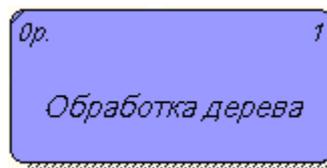


Рисунок 65 - Изображение работы

Внешние сущности изображают входы в систему и/или выходы из системы. Внешние сущности изображаются в виде прямоугольника с тенью и обычно располагаются по краям диаграммы (рис. 66). Одна внешняя сущность может быть использована многократно на одной или нескольких

диаграммах. Обычно такой прием применяют, чтобы не рисовать слишком длинных и запутанных стрелок.

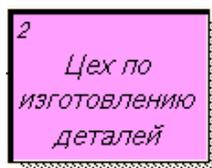


Рисунок 66 - Внешняя сущность

Стрелки (потоки данных) описывают движение объектов из одной части системы в другую. Поскольку в DFD каждая сторона работы не имеет четкого назначения, как в IDEF0, стрелки могут подходить и выходить из любой грани прямоугольника работы. В DFD также применяются двунаправленные стрелки для описания диалогов типа команды-ответа между работами, между работой и внешней сущностью и между внешними сущностями (рис. 67).

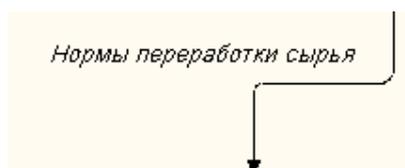


Рисунок 67 - Потоки данных (стрелки)

Хранилище данных изображают объекты в покое. В материальных системах хранилища данных изображаются там, где объекты ожидают обработки, например в очереди. В системах обработки информации хранилища данных являются механизмом, который позволяет сохранить данные для последующих процессов (рис. 68).



Рисунки 68 - Хранилище данных

В отличие от стрелок IDEF0, которые представляют собой жесткие взаимосвязи, стрелки DFD показывают, как объекты (включая данные) двигаются от одной работы к другой. Это представление потоков совместно с хранилищами данных и внешними сущностями делает модели DFD более

похожими на физические характеристики системы - движение объектов (*data flow*), хранение объектов (*data stores*), поставка и распространение объектов (*external entities*) (рис. 69).

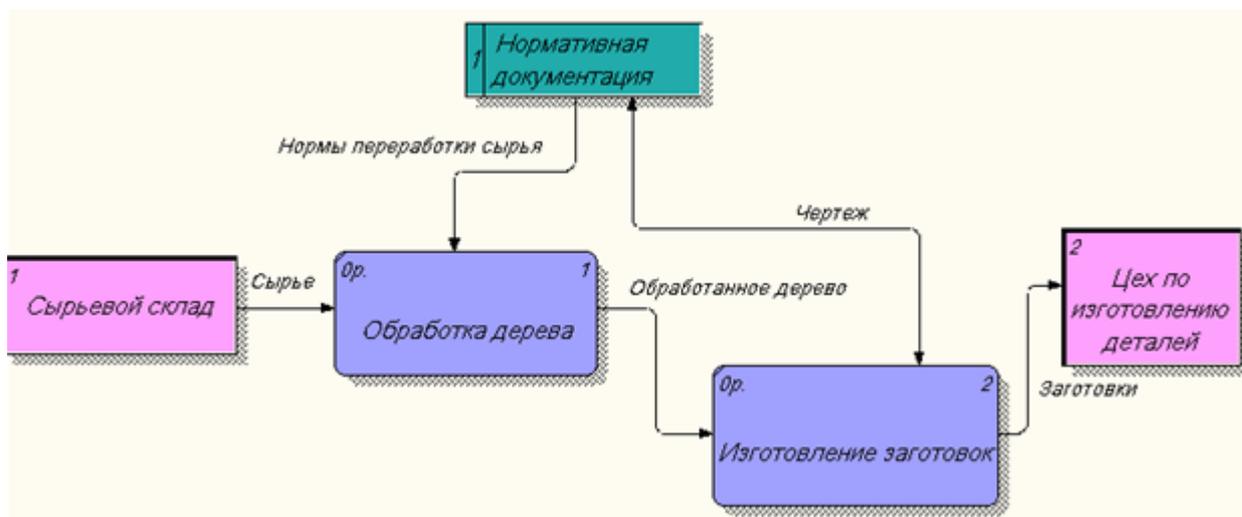


Рисунок 69 - Пример диаграммы DFD

Практическое задание

Построение модели рассмотрим на примере бизнес-процесса "Переработка сырья".



Задание 26. Создание диаграммы DFD.

1. Откройте файл **Lab4.bp1**, сохраненный на предыдущем уроке.
2. Выберите инструмент  и в процессе декомпозиции работы «Переработка сырья» в диалоге **Activity Box Count** «кликните» по радиокнопке **DFD**.
3. В диалоге **Activity Box Count** выберите количество работ – 2 (рис. 70).

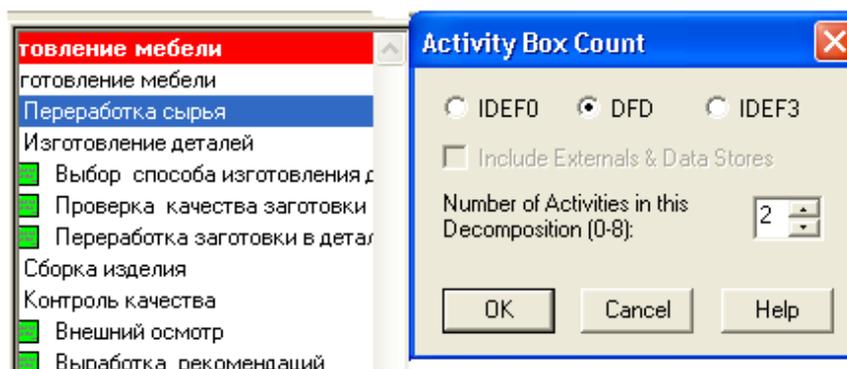


Рисунок 70 - Диалог Activity Box Count

4. Щелкните по кнопке ОК и внесите в новую диаграмму имена работ: «Обработка дерева», «Изготовление заготовок».



Задание 27. Изображение внешних сущностей.

Выполните следующие действия:

1. Используя кнопку , внесите внешние сущности (внешние ссылки).
2. В появившемся диалоговом окне введите название сущностей: «Сырьевой склад», «Цех по изготовлению деталей» (рис. 71) и нажмите ОК.

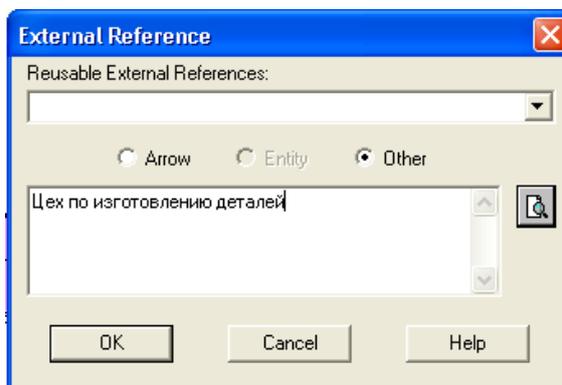


Рисунок 71 - Диалоговое окно внешней сущности

После выполнения задания у вас должна получиться следующая диаграмма (рис. 72).

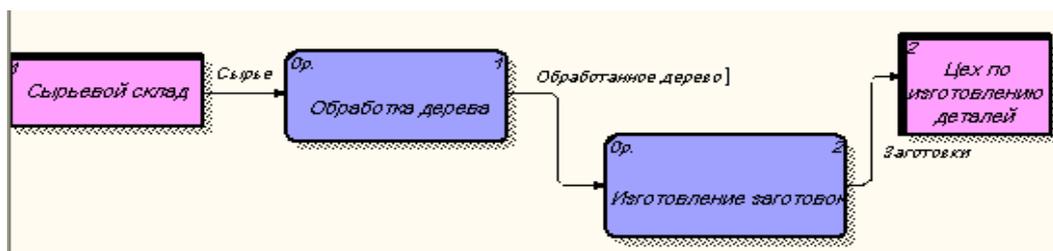


Рисунок 72 - Изображение внешних сущностей



Задание 28. Построение хранилищ.

1. Используя кнопку , внесите хранилище данных: «Нормативная документация» (рис. 73);
2. Удалите граничные стрелки с диаграммы.

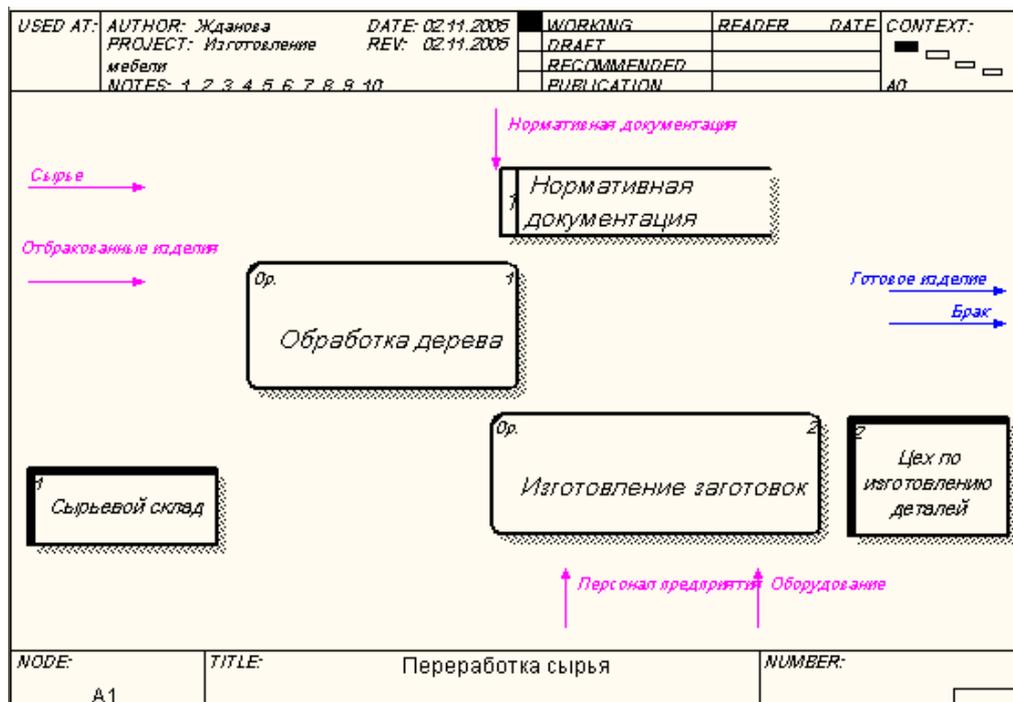


Рисунок 73 - Диаграмма DFD



Задание 29. Создание внутренних ссылок.

- Используя инструмент , создайте внутренние ссылки: «Сырье», «Обработанное дерево»; «Заготовки», «Нормы переработки сырья», «Чертеж».
- Стрелку «Чертеж» необходимо сделать двунаправленной, для этого щелкните правой кнопкой по стрелке, выберите в контекстном меню пункт **Style**.
- Выберите в диалоговом окне **Arrow Properties** опцию **Bidirectional** (рис 74).

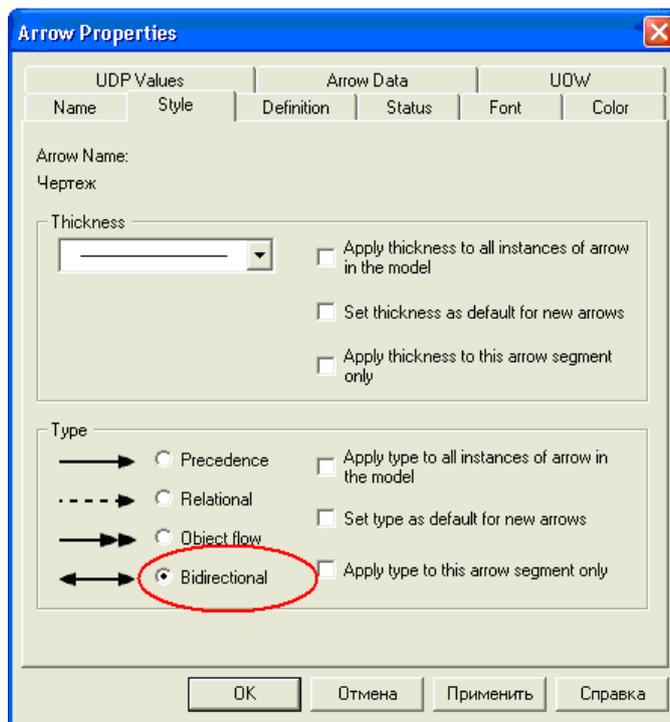


Рисунок 74 - Диалоговое окно выбора стиля стрелки

4. У вас должна получиться диаграмма.



Задание 30. Детализация процесса «Изготовление заготовок».

Самостоятельно проведите детализацию процесса "Изготовление заготовок" в соответствии с рисунком (рис. 75).

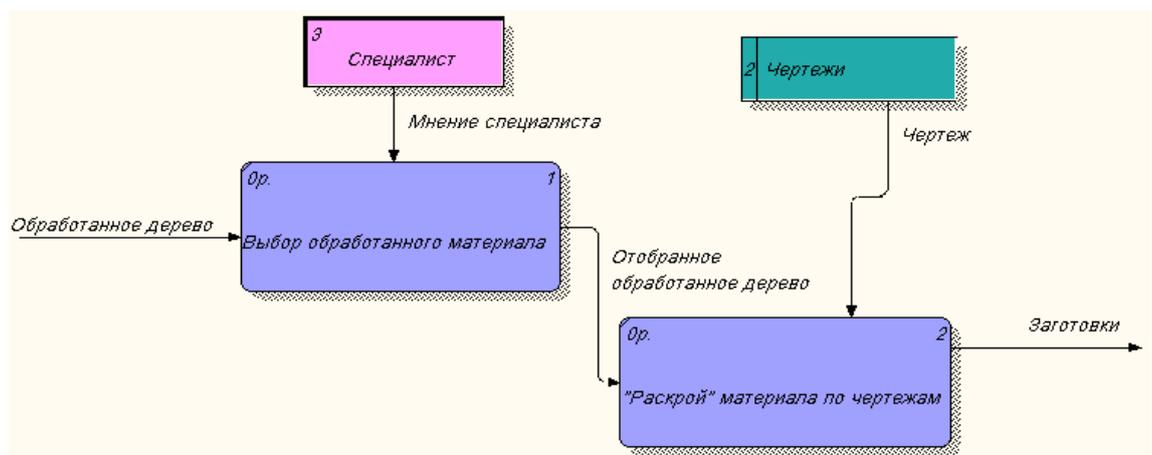


Рисунок 75 - Детализация процесса «Изготовление заготовок»

 После того, как эта работа будет проделана, пригласите преподавателя и продемонстрируйте ему результат.



Задание 31. Сохранение полученной диаграммы.

Сохраните полученную диаграмму.

1. В меню **File** выберите **Save as**.
2. Укажите путь к своей папке и имя файла **Lab5.bp1**.
3. Нажмите ОК.

Контрольные вопросы

1. Для чего служит DFD - диаграмма?
2. В чем отличие DFD - диаграммы от IDFE0?
3. Какой инструмент используется для построения внешних сущностей?
4. Каким инструментом можно построить Хранилище данных?
5. Какие графические элементы используются для обозначения на диаграмме Работы, Поток данных, Хранилищ данных?

Тестовое задание

1. **DFD - это:**
 - a) диаграмма бизнес - процесса;
 - b) диаграмма потока данных;
 - c) контекстная диаграмма;
 - d) функциональная диаграмма;
 - e) диаграмма сущность - связь.
2. **Какие объекты описываются на диаграмме потоков данных:**
 - a) функции (работы);
 - b) хранилища данных;
 - c) внешние объекты;
 - d) внешние ссылки;
 - e) все ответы правильные.
3. **Что показывают диаграммы потоков данных?**
 - a) внешние источники и приемники данных;
 - b) потоки данных;
 - c) хранилища (накопители) данных;
 - d) все ответы правильные;
 - e) правильного ответа нет.

4. Какой инструмент нужно выбрать для создания диаграммы DFD?

- a)  ;
- b)  ;
- c)  ;
- d)  ;
- e)  .

5. Укажите номер рисунка, обозначающий на диаграмме Хранилище данных.

- a) 
- b) 
- c) 

6. Каким инструментом можно обозначить в диаграмме DFD Внешнюю сущность?

- a)  ;
- b)  ;
- c)  ;
- d)  ;
- e)  .

 После того, как эта работа будет проделана, пригласите преподавателя и продемонстрируйте ему результат.

Контрольное задание

Продолжите выполнение контрольного задания.

1. Откройте файл **Kr4.bp1**.
2. Постройте диаграмму **DFD**.
3. Детализируйте диаграмму **DFD**.
4. Сохраните файл в своей папке с именем **Kr5.bp1**.

 После того, как эта работа будет проделана, пригласите преподавателя и продемонстрируйте ему результат.

Лабораторная работа № 6. Создание диаграммы IDEF3

С помощью этой лабораторной работы Вы сможете:

- освоить принципы построения диаграммы IDEF3;
- научиться устанавливать связи между работами;
- освоить правила создания перекрестков.

Теоретические сведения

Диаграммы IDEF3

 Наличие в диаграммах DFD элементов для описания источников, приемников и хранилищ данных позволяет более эффективно и наглядно описать процесс документооборота.

Однако для *описания логики взаимодействия информационных потоков* более подходит **IDEF3**, называемая также **workflow diagramming** - методологией моделирования, использующая графическое описание информационных потоков, взаимоотношений между процессами обработки информации и объектов, являющихся частью этих процессов.

Диаграммы **Workflow** могут быть использованы в моделировании бизнес-процессов для анализа завершенности процедур обработки информации. С их помощью можно описывать сценарии действий сотрудников организации, например, последовательность обработки заказа или события, которые необходимо обработать за конечное время. Каждый сценарий сопровождается описанием процесса и может быть использован для документирования каждой функции.

IDEF3 - это метод, имеющий основной целью дать возможность аналитикам *описать ситуацию, когда процессы выполняются в определенной последовательности, а также описать объекты, участвующие совместно в одном процессе.*

Каждая работа в **IDEF3** описывает какой-либо сценарий бизнес-процесса и может являться составляющей другой работы. Поскольку сценарий описывает цель и рамки модели, важно, чтобы работы именовались отглагольным существительным, обозначающим процесс действия, или именным словосочетанием, содержащим такое существительное.

Точка зрения на модель должна быть задокументирована. Обычно это точка зрения человека, ответственного за работу в целом. Также необходимо задокументировать цель модели - те вопросы, на которые призвана ответить модель.

Диаграмма является основной единицей описания в **IDEF3**. Важно правильно построить диаграммы, поскольку они предназначены для чтения другими людьми (а не только автором).

Единицы работы – Unit of Work (UOW), также называемые работами (activity), являются центральными компонентами модели. В **IDEF3** работы изображаются *прямоугольниками с прямыми углами* (рис. 76) и имеют **имя**, выраженное отглагольным существительным, *обозначающим процесс действия*, **одиночным** или в составе словосочетания, и **номер** (идентификатор); другое имя существительное в составе того же словосочетания, зависимое от отглагольного существительного, обычно отображает основной выход (результат) работы (например, "Изготовление изделия").

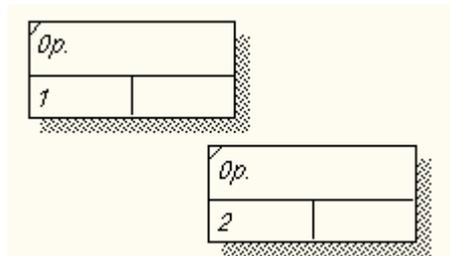


Рисунок 76 - Обозначение работы в диаграмме IDEF3

Связи показывают взаимоотношения работ. Все связи в **IDEF3** однонаправлены и могут быть направлены куда угодно, но обычно диаграммы IDEF3 стараются построить так, чтобы *связи были направлены слева направо*. В IDEF3 различают три типа стрелок, изображающих связи, стиль которых устанавливается во вкладке **Style** (рис. 77) диалога **Arrow Properties** (пункт контекстного меню **Style**).

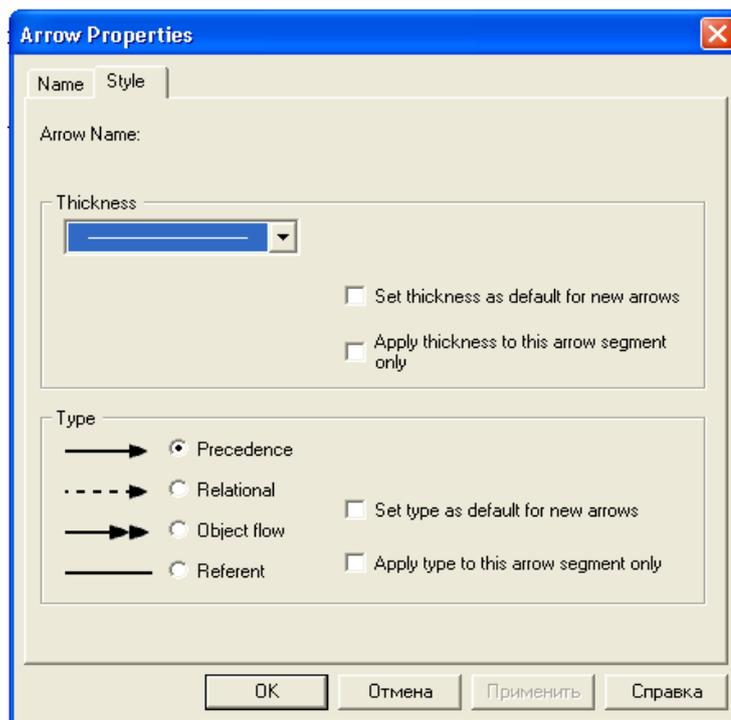


Рисунок 77 - Вкладка Style диалога Arrow Properties

Старшая (Precedence) стрелка  Precedence - сплошная линия, связывающая единицы работ (UOW). Рисуеться слева направо или сверху вниз. Показывает, что работа-источник должна закончиться прежде, чем работа-цель начнется.

Стрелка отношения (Relational)  Relational - пунктирная линия, используемая для изображения связей между единицами работ (UOW), а также между единицами работ и объектами ссылок.

Потоки объектов (Object Flow)  Object flow - стрелка с двумя наконечниками, применяется для описания того факта, что объект используется в двух или более единицах работы, например, когда объект порождается в одной работе и используется в другой.

Старшая связь показывает, что работа-источник заканчивается ранее, чем начинается работа-цель. Часто результатом работы-источника становится объект, необходимый для запуска работы-цели. В этом случае стрелку, обозначающую объект, изображают с двойным наконечником. Имя стрелки должно ясно идентифицировать отображаемый объект. Поток объектов имеет ту же семантику, что и старшая стрелка.

Отношение показывает, что стрелка является альтернативой старшей стрелке или потоку объектов в смысле задания последовательности выполнения работ - работа-источник не обязательно должна закончиться прежде, чем работа-цель начнется. Более того, работа-цель может закончиться прежде, чем закончится работа-источник (рис. 78).

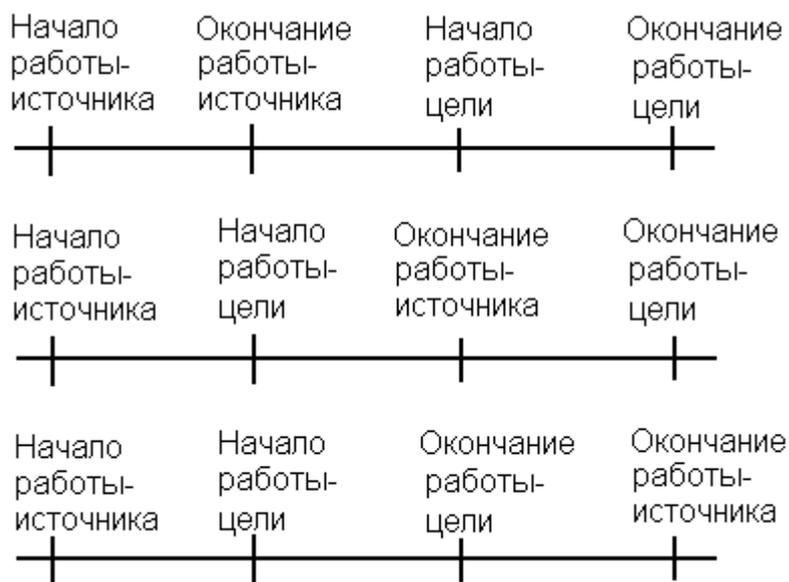


Рисунок 78 - Временная диаграмма выполнения работ

Перекрестки (Junction). Окончание одной работы может служить сигналом к началу нескольких работ, или же одна работа для своего запуска может ожидать окончания нескольких работ. Перекрестки используются для отображения логики взаимодействия стрелок при слиянии и разветвлении или для отображения множества событий, которые могут или должны быть завершены перед началом следующей работы.

Различают *перекрестки для слияния (Fan-in Junction)* и разветвления (**Fan-in Junction**) стрелок. Перекресток не может использоваться одновременно для слияния и разветвления.

Для внесения перекрестка служит кнопка  в палитре инструментов. В диалоге **Junction Type Editor** нужно будет указать тип перекрестка (рис. 79).

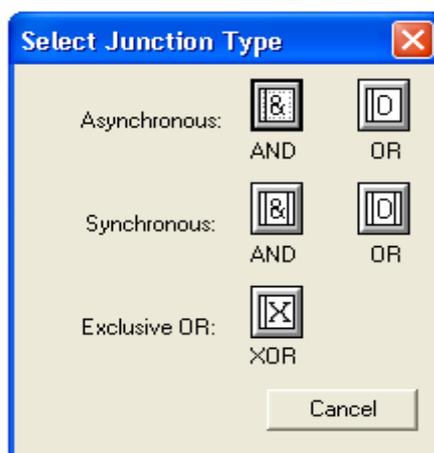


Рисунок 79 - Типы перекрестков

Смысл каждого типа приведен в таблице 5.

Таблица 5 - Типы перекрестков

Обозначение	Наименование	Смысл в случае слияния стрелок Fan-in Junction	Смысл в случае разветвления стрелок Fan-in Junction
Asynchronous:  AND	Асинхронное «И» (Asynchronous AND)	Все предшествующие процессы должны быть завершены	Все следующие процессы должны быть запущены
Synchronous:  AND	Синхронное «И» (Synchronous AND)	Все предшествующие процессы завершены одновременно	Все следующие процессы запускаются одновременно
Asynchronous:  OR	Асинхронное «ИЛИ» (Asynchronous OR)	Один или несколько предшествующих процессов должны быть завершены	Один или несколько следующих процессов должны быть запущены
Synchronous:  OR	Синхронное «ИЛИ» (Synchronous OR)	Один или несколько предшествующих процессов завершены одновременно	Один или несколько следующих процессов запускаются одновременно
Exclusive OR:  XOR	Исключающее «ИЛИ» XOR (Exclusive OR)	Только один предшествующий процесс завершен	Только один следующий процесс запускается

Все перекрестки на диаграмме нумеруются, каждый номер имеет префикс **J** (рис. 80).

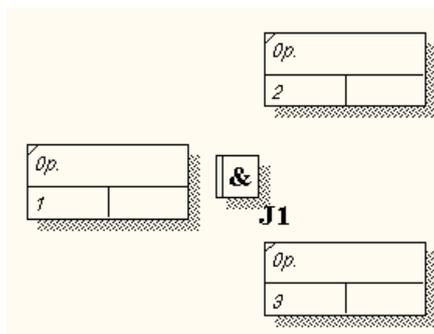


Рисунок 80 - Обозначение нумерации перекрестка

Можно редактировать свойства перекрестка (рис 81) при помощи диалога **Junction Properties**, который вызывается из контекстного меню.

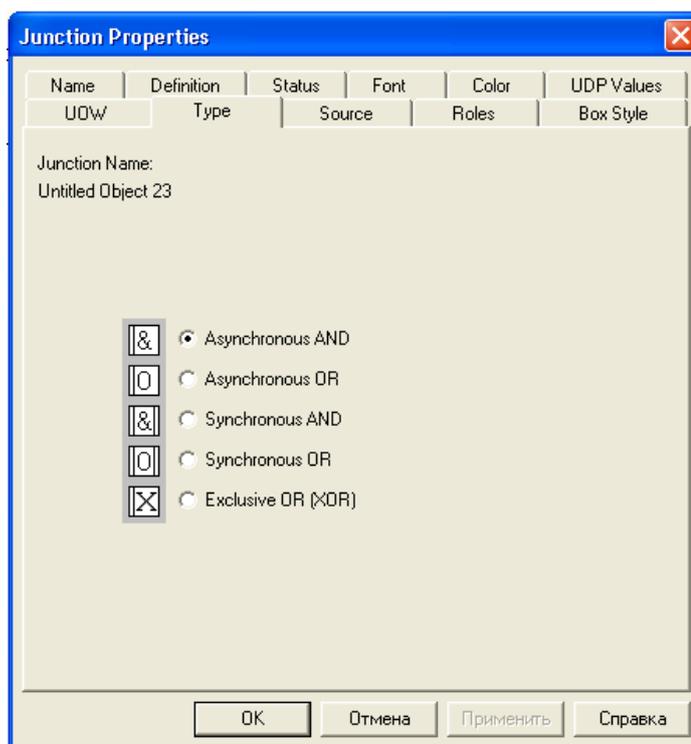


Рисунок 81 - Диалоговое окно свойств перекрестков

В отличие от IDEF0 и DFD в IDEF3 стрелки могут сливаться и разветвляться только через перекрестки.

Правила создания перекрестков. На одной диаграмме IDEF3 может быть создано несколько перекрестков различных типов. Определенные сочетания перекрестков для слияния и разветвления могут приводить к

логическим несоответствиям. Чтобы избежать конфликтов, необходимо соблюдать следующие правила:

1. Каждому перекрестку для слияния должен предшествовать перекресток для разветвления.

2. Перекресток для слияния «И» не может следовать за перекрестком для разветвления типа *синхронного* или *асинхронного* «ИЛИ». Действительно, после работы 1 может запускаться только одна работа — 2 или 3, а для запуска работы 4 требуется окончание обеих работ - 2 и 3. Такой сценарий не может реализоваться (рис. 82).

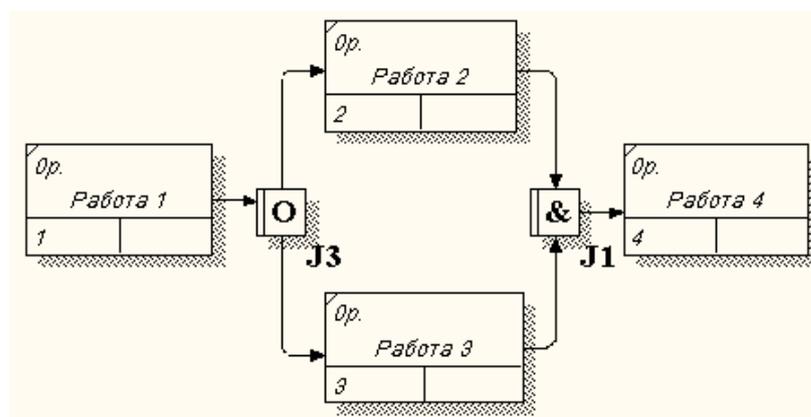


Рисунок 82 - Неверное размещение перекрестков. Перекресток для слияния «И» не может следовать за перекрестком для разветвления «ИЛИ»

3. Перекресток для слияния «И» не может следовать за перекрестком для разветвления типа *исключающего* «ИЛИ» (рис. 83).

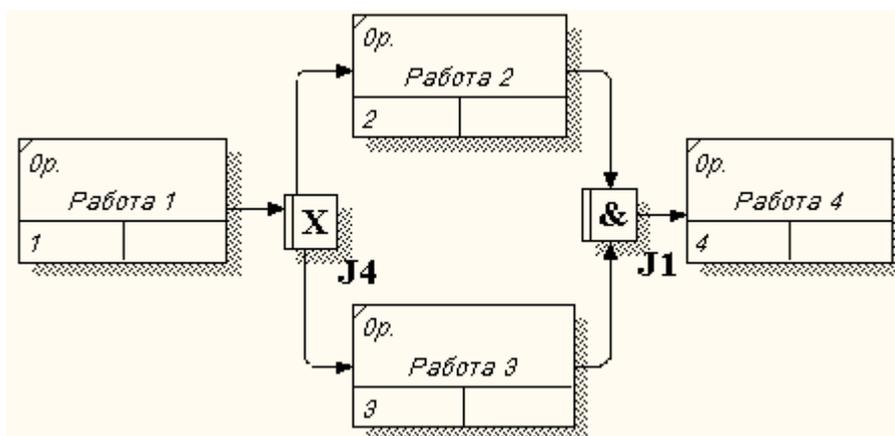


Рисунок 83 - Неверное размещение перекрестков. Перекресток для слияния «И» не может следовать за перекрестком для разветвления типа *исключающего* «ИЛИ»

4. Перекресток для слияния типа *исключающего «ИЛИ»* не может следовать за перекрестком для разветвления типа «И» (рис. 84). Здесь после завершения работы 1 запускаются обе работы - 2 и 3, а для запуска работы 4 требуется, чтобы завершилась одна и только одна работа - или 2, или 3.

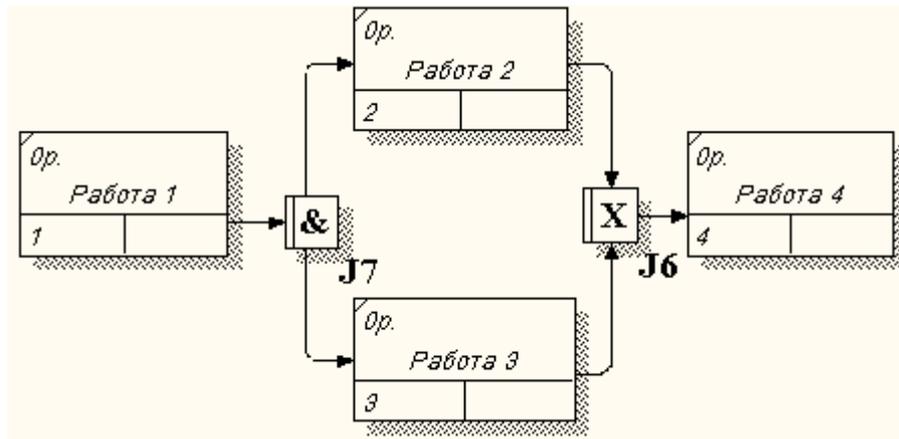


Рисунок 84 - Неверное размещение перекрестков. Перекресток для слияния типа *исключающего «ИЛИ»* не может следовать за перекрестком для разветвления типа «И»

5. Перекресток, имеющий одну стрелку на одной стороне, должен иметь более одной стрелки на другой.

Практическое задание

Построение модели рассмотрим на примере бизнес-процесса "Сборка изделия".



Задание 32. Создание диаграммы IDEF3.

1. Откройте файл **Lab5.bp1**, сохраненный на предыдущем уроке.
2. Перейдите на диаграмму А3 и, выбрав инструмент , декомпозируйте работу «Сборка изделия».
3. В диалоге **Activity Box Count** установите число работ 4 и нотацию **IDEF3** (рис. 85).

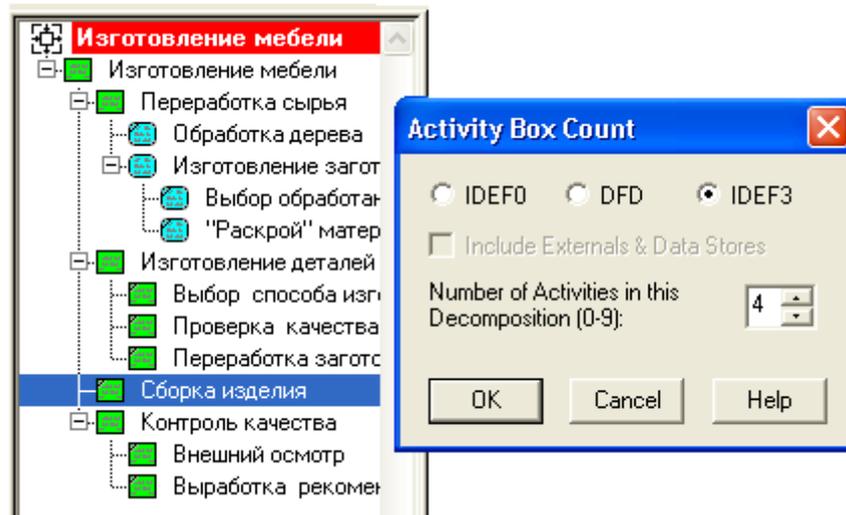


Рисунок 85 - Выбор нотации IDEF3 в диалоге Activity Box Count

Возникает диаграмма **IDEF3**, содержащая работы (**UOW**).

4. Правой кнопкой мыши щелкните по работе, выберите в контекстном меню **Name** и внесите имя работы «Подготовка компонентов».
5. Во вкладке **Definition** внесите определение «Подготавливаются все компоненты корпусной мебели согласно спецификации заказа» (рис. 86).

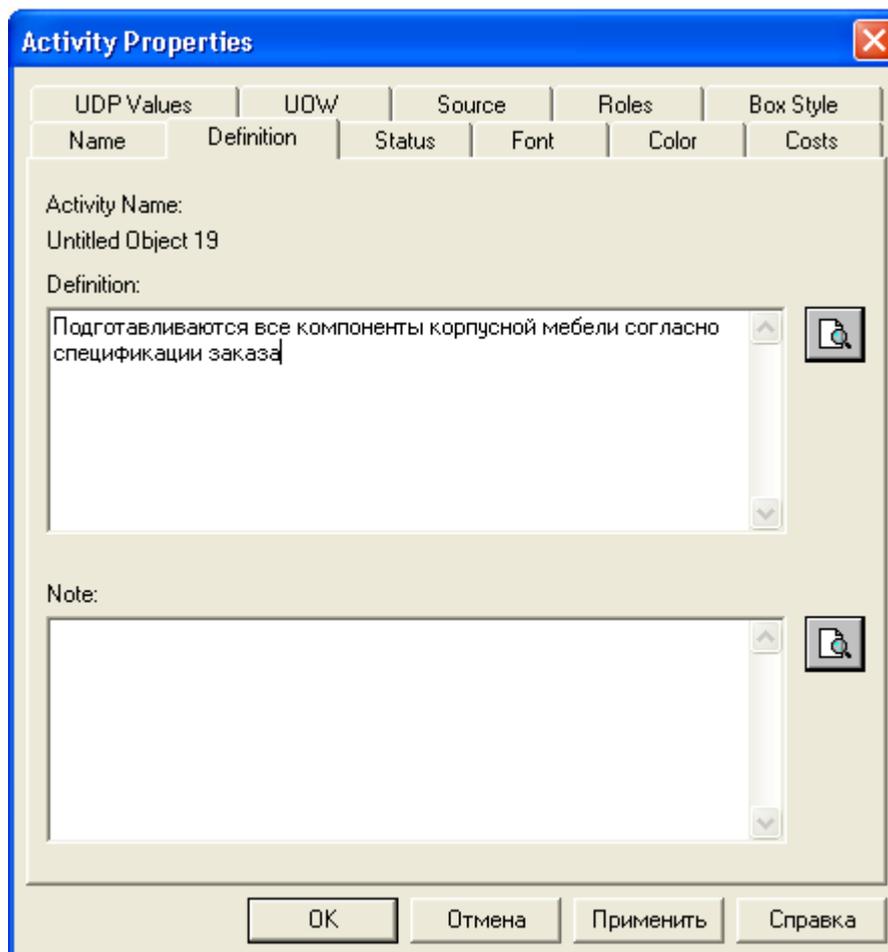


Рисунок 86 - Диалоговое окно свойств работы

6. Во вкладку **UOW**, внесите свойства работы (таблица 6).

Таблица 6 - Свойства UOW

<i>Tun</i>	<i>Использование</i>
Name	Подготовка деталей изделия
Definition	Подготавливаются все детали изделия согласно спецификации заказа
Objects	Детали: основания, боковые стенки, полки, дверцы, зеркала, стекла, фурнитура
Constrains	Установка зеркал возможна только после полной сборки шкафа

7. Внесите в диаграмму еще 4 работы (кнопка ).

8. Внесите имена следующих работ: «Сборка основания», «Установка боковых стенок», «Установка задней стенки», «Установка зеркала», «Размещение полочек», «Установка дверок», «Прикручивание фурнитуры» (рис. 87).

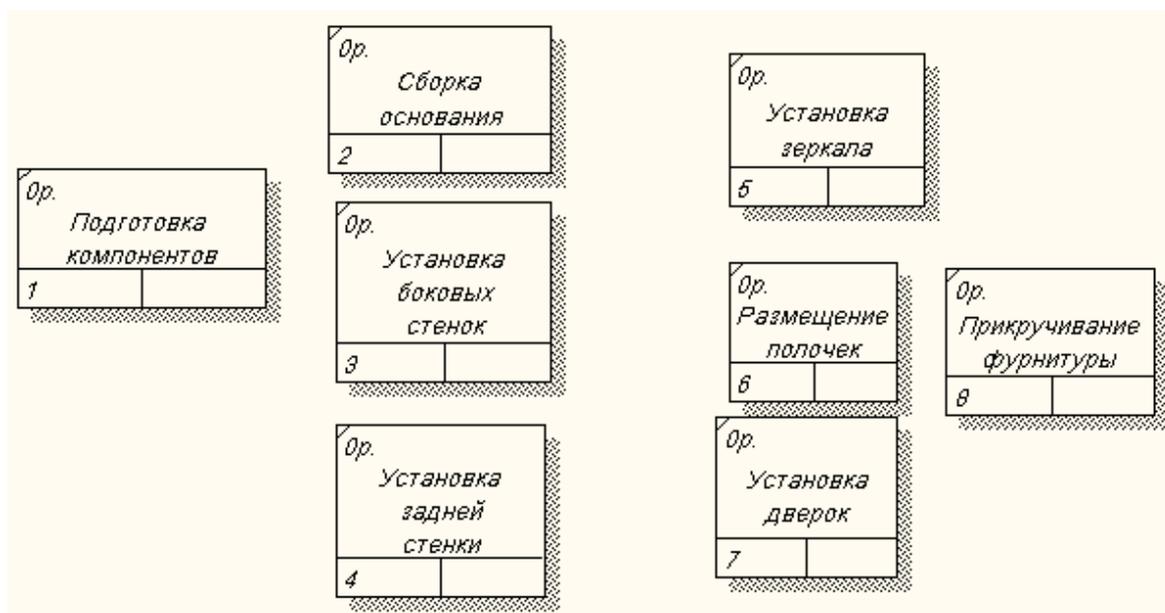


Рисунок 87 - Фрагмент диаграммы IDEF3



Задание 33. Добавление в диаграмму IDEF3 объекта ссылки.

1. С помощью кнопки  (добавить в диаграмму объект ссылки - *Referent*), расположенной в палитре инструментов, создайте объект ссылки.
2. Внесите имя объекта внешней ссылки «Компоненты».

3. Измените стиль стрелки на **Referent** (рис. 88).

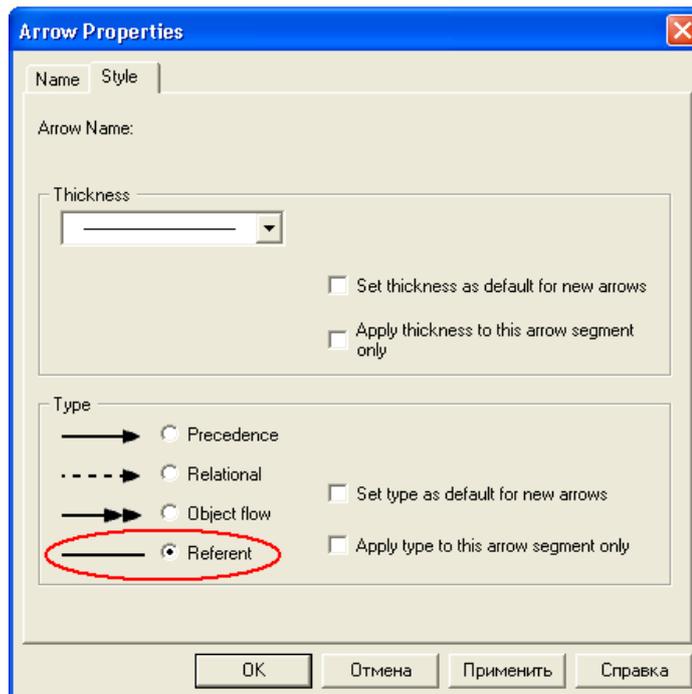


Рисунок 88 - Диалоговое окно определения стиля стрелок

Объект ссылки в **IDEF3** выражает некую идею, концепцию или данные, которые нельзя связать со стрелкой, перекрестком или работой.



Задание 34. Связывание работ с помощью стрелок.

1. Свяжите стрелкой работы «Подготовка компонентов» (выход) и «Сборка основания».

2. Измените стиль стрелки на **Object Flow** (рис. 89).



В **IDEF3** имя стрелки может отсутствовать, хотя VPwin показывает отсутствие имени как ошибку.

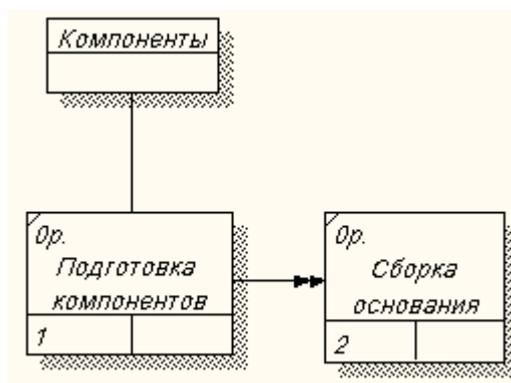


Рисунок 89 - Результат создания UOW и объекта ссылки

3. Так как работы «Установка боковых стенок», «Установка задней стенки», «Установка зеркала» должны следовать только друг за другом, соедините их стрелкой **Precedence** , она показывает, что работа-источник должна закончиться прежде, чем работа-цель начнется (рис. 90).

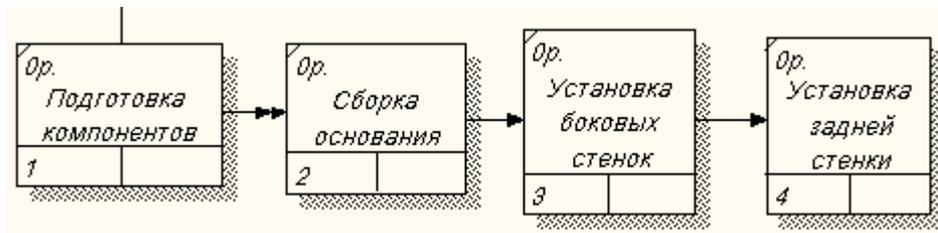


Рисунок 90 - Результат связывания последовательно выполняемых работ



Задание 35. Установление перекрестков.

1. С помощью кнопки  на палитре инструментов внесите один перекресток типа *асинхронного* «ИЛИ» (один или несколько следующих процессов должны быть запущены) и свяжите работы с перекрестком, как показано на рис. 91 (это случай разветвления стрелок **Fan-in Junction**).

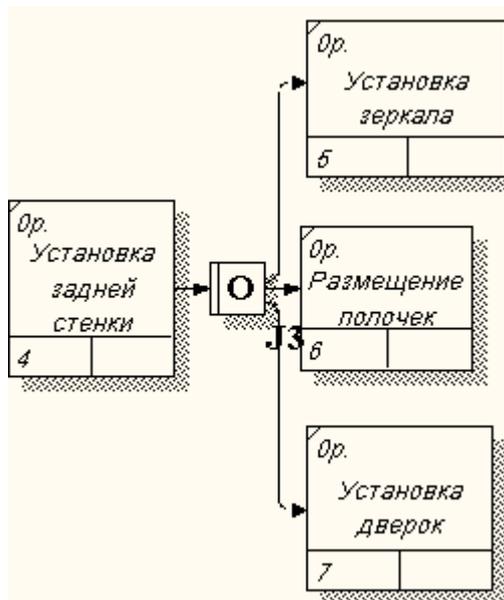


Рисунок 91 - Результат создания перекрестка

2. Правой кнопкой щелкните по перекрестку для разветвления, выберите **Name** и внесите имя «Компоненты, требуемые в спецификации заказа».

3. Самостоятельно установите перекресток для слияния стрелок.
4. Проверьте себя (рис.92).

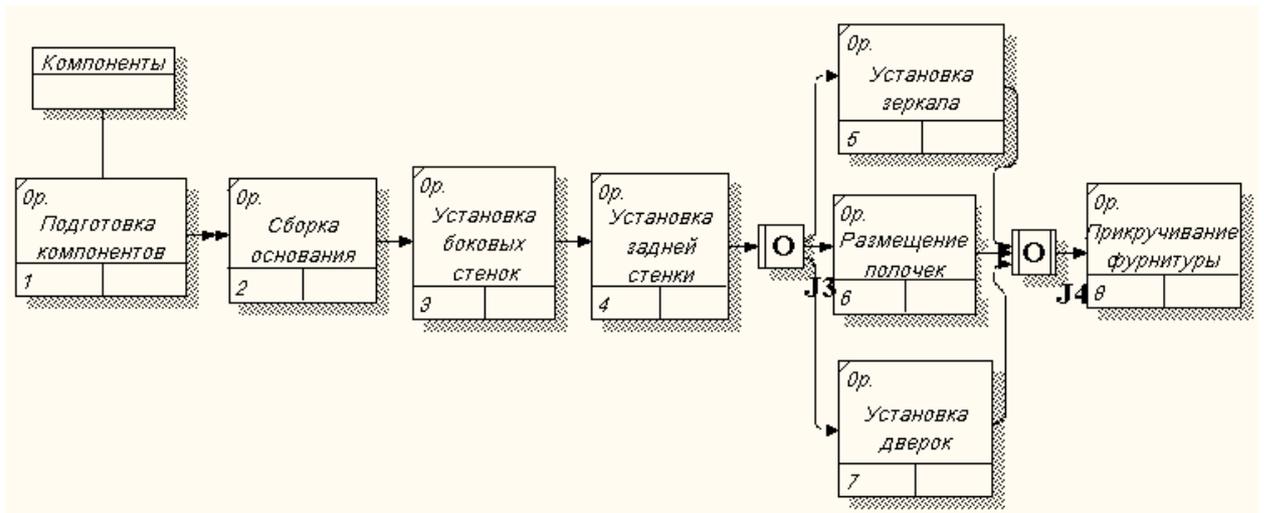


Рисунок 92 - Диаграмма IDEF3

Примечание. Так как работа «Прикручивание фурнитуры» должна производиться после окончания всех предыдущих работ, то выбираемый тип перекрестка - *синхронное* «ИЛИ»



 После того, как эта работа будет проделана, пригласите преподавателя и продемонстрируйте ему результат.



Задание 36. Сохранение полученной диаграммы.

Сохраните полученную диаграмму.

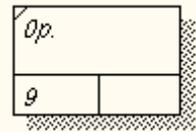
1. В меню **File** выберите **Save as**.
2. Укажите путь к своей папке и имя файла **Lab6.bp1**.
3. Нажмите ОК.

Контрольные вопросы

1. Для чего строится диаграмма IDEF3?
2. Чем диаграмма IDEF3 отличается от диаграммы IDEF0?
3. Как графически обозначается работа в диаграмме IDEF3?
4. С какой целью между работами устанавливаются перекрестки?
5. Какие типы перекрестков вам знакомы?

Тестовое задание

1. В какой нотации используется следующее графическое



изображение работы:

- a) DFD;
- b) IDEF3;
- c) IDEF0;
- d) правильные ответы b), c);
- e) правильные ответы a), c).

2. Укажите неверное название работы:

- a) «Обслуживание клиента»;
- b) «Оформление заказа»;
- c) «Качественная фурнитура»;
- d) «Обработка дерева»;
- e) правильного ответа нет.

3. С помощью какого инструмента можно добавить в диаграмму объект-ссылку?

- a) ;
- b) ;
- c) ;
- d) ;
- e) .

4. Что показывает стрелка соединяющая работы?



- a) стрелка показывает, что работа-источник должна закончиться прежде, чем работа-цель начнется;
- b) стрелка показывает связь между единицами работ и объектами ссылок;
- c) стрелка показывает, что объект используется в двух или более единицах работы;
- d) все ответы правильные.

5. С помощью какого инструмента можно установить между работами перекресток?

- a) ;
- b) ;
- c) ;
- d) .

e) .

6. Распределите перекрестки в таблице, в соответствии с их смыслом.

Обозначение	Смысл в случае слияния стрелок Fan-in Junction	Смысл в случае разветвления стрелок Fan-out Junction
	Все предшествующие процессы должны быть завершены	Все следующие процессы должны быть запущены
	Один или несколько предшествующих процессов должны быть завершены	Один или несколько следующих процессов должны быть запущены
	Все предшествующие процессы завершены одновременно	Все следующие процессы запускаются одновременно
	Один или несколько предшествующих процессов завершены одновременно	Один или несколько следующих процессов запускаются одновременно

 Asynchronous: **AND**
 Exclusive OR: **XOR**
 Asynchronous: **OR**
 Synchronous: **OR**
 Synchronous: **AND**

 После того, как эта работа будет проделана, пригласите преподавателя и продемонстрируйте ему результат.

Контрольное задание

Продолжите выполнение контрольного задания.

1. Откройте файл **Kr5.bp1**.
2. Постройте диаграмму **IDEF3**, используя перекрестки, добавьте в диаграмму объект ссылки.
3. Сохраните файл в своей папке с именем **Kr6.bp1**.

 После того, как эта работа будет проделана, пригласите преподавателя и продемонстрируйте ему результат.

Лабораторная работа № 7. Стоимостный анализ

С помощью этой лабораторной работы Вы сможете:

- освоить последовательность и правила определения параметров стоимостного анализа;
- научиться составлять отчет о стоимости бизнес-процесса.

Теоретические сведения

Стоимостный анализ (ABC)

 Для того чтобы определить качество созданной модели с точки зрения эффективности бизнес-процессов, необходима система метрики, т. е. качество следует оценивать количественно.

ВРwin предоставляет аналитику два инструмента для оценки модели - стоимостный анализ, основанный на работах (**Activity Based Costing, ABC**), и свойства, определяемые пользователем (**User Defined Properties, UDP**).

ABC (стоимостный анализ) является широко распространенной методикой, используемой международными корпорациями и государственными организациями (в том числе Департаментом обороны США) для идентификации истинных движителей затрат в организации. Именно эту методику мы рассмотрим в лабораторной работе.

Стоимостный анализ представляет собой *соглашение об учете, используемое для сбора затрат, связанных с работами, с целью определить общую стоимость процесса.*

Стоимостный анализ основан на модели работ, потому что количественная оценка невозможна без детального понимания функциональности предприятия.

Обычно **ABC** применяется для того, чтобы понять происхождение выходных затрат и облегчить выбор нужной модели работ при реорганизации деятельности предприятия. С помощью стоимостного анализа можно решить такие задачи, как:

- 1) определение действительной стоимости производства продукта,
- 2) определение действительной стоимости поддержки клиента,
- 3) идентификация работ, которые стоят больше всего (те, которые должны быть улучшены в первую очередь),

4) обеспечение менеджеров финансовой мерой предлагаемых изменений и др.

АВС может проводиться только тогда, когда *модель работы последовательная* (следует синтаксическим правилам **IDEF0**), *корректная* (отражает бизнес), *полная* (охватывает всю рассматриваемую область) и *стабильная* (проходит цикл экспертизы без изменений), другими словами, *когда создание модели работы закончено*.

АВС включает следующие основные понятия:

- **объект затрат** - причина, по которой работа выполняется; обычно, основной выход работы.

Стоимость работ есть суммарная стоимость объектов затрат («Готовое изделие», рис. 93)

- **движитель затрат** — характеристики входов и управлений работы ("Сырье", «Нормативная документация», рис. 93), которые влияют на то, как выполняется и как долго длится работа;

- **центры затрат**, которые можно трактовать как статьи расхода.

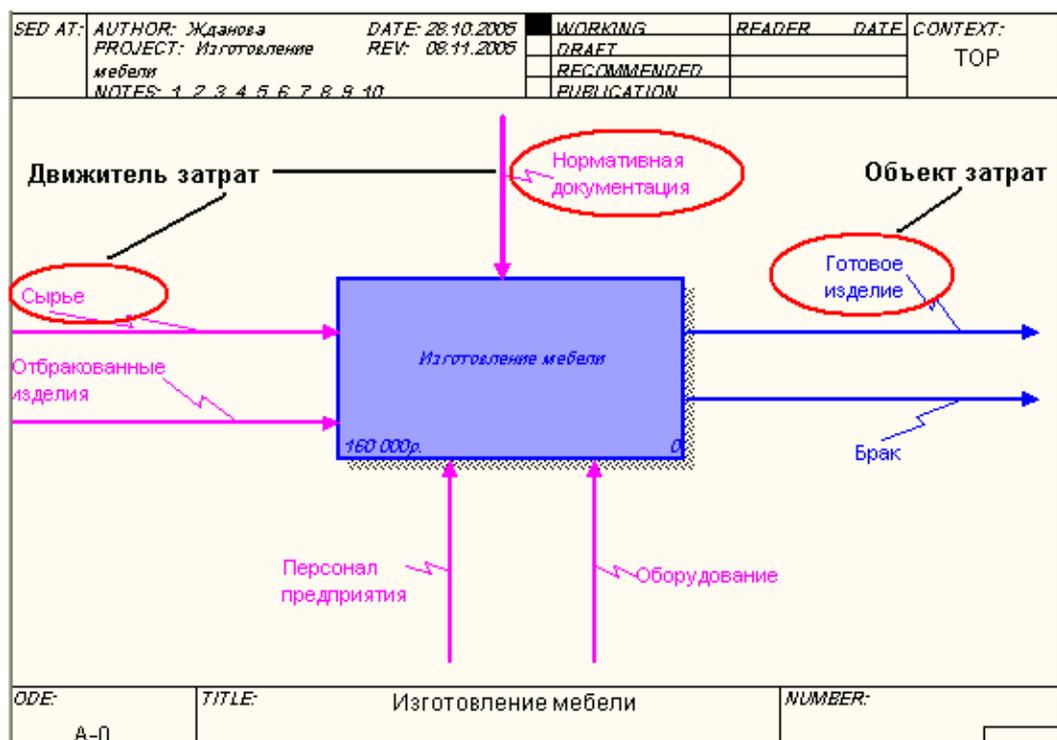


Рисунок 93 - Иллюстрация терминов АВС

При проведении стоимостного анализа в VPwin сначала задаются единицы измерения времени и денег.

Общие затраты по работе рассчитываются как сумма по всем центрам затрат. При вычислении затрат вышестоящей (родительской) работы сначала вычисляется произведение затрат дочерней работы на *частоту* работы (число раз, которое работа выполняется в рамках проведения родительской работы), затем результаты складываются. Если во всех работах модели включен режим **Compute from Decompositions**, подобные вычисления автоматически проводятся по всей иерархии работ снизу вверх (рис. 94).

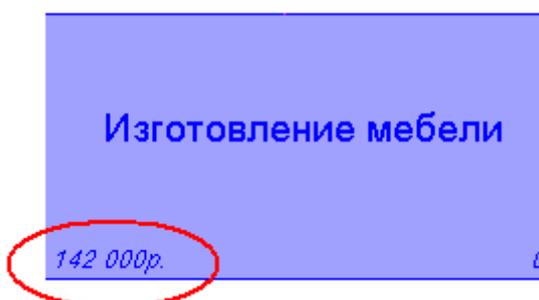


Рисунок 94 - Общие затраты на работу «Изготовление мебели»

Этот достаточно упрощенный принцип подсчета справедлив, если работы выполняются последовательно. Встроенные возможности VPwin позволяют разрабатывать упрощенные модели стоимости, которые тем не менее оказываются чрезвычайно полезными для предварительной оценки затрат. Если схема выполнения более сложная (например, работы производятся альтернативно), можно отказаться от подсчета и задать итоговые суммы для каждой работы вручную.

Практическое задание



Задание 37. Настройка параметров стоимостного анализа.

1. Откройте файл **Lab6.bp1**, сохраненный на предыдущем уроке.
2. В диалоговом окне **Model Properties (Model - Model Properties)** во вкладке **ABC** установите единицы измерения денег и времени – рубли и часы (рис.95).

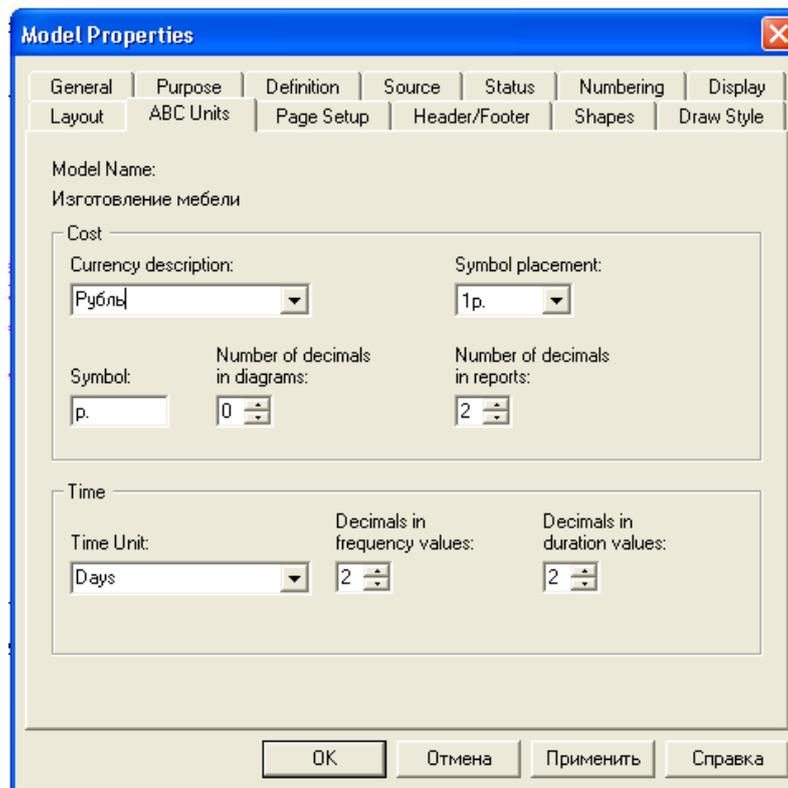


Рисунок 95 - Вкладка ABC Unit диалога Model Properties

3. Перейдите в **Dictionary – Cost Center** и в диалоге **Cost Center Dictionary** внесите название и определение центров затрат (табл. 7).

Таблица 7 - Центры затрат ABC

Центр затрат	Определение
Управление	Затраты на управление, связанные с составлением графика работ, формированием комплектов мебели, контролем над сборкой и проверкой качества изделий
Рабочая сила	Затраты на оплату рабочих, занятых изготовлением изделий
Компоненты	Затраты на закупку компонентов

4. Для отображения стоимости каждой работы в нижнем левом углу прямоугольника перейдите в меню **Model - Model Properties** и во вкладку **Display** диалога **Model Properties** включите опцию **ABC Data** (рис. 96).

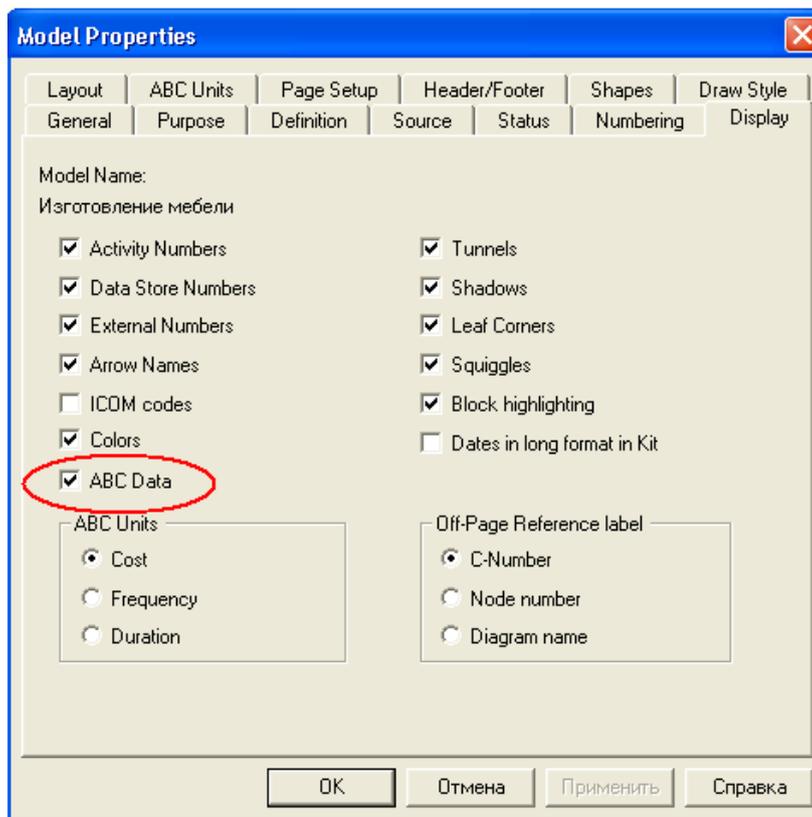


Рисунок 96 - Вкладка Display диалога Model Properties

5. Для изображения стоимости, частоты или продолжительности работы переключите радиокнопки в группе **ABC Units** (рис. 97).

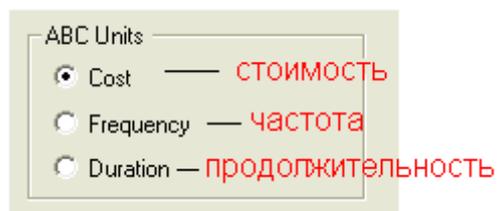


Рисунок 97 - Назначение радиокнопок в группе ABC Units

6. Для назначения стоимости работе «Переработка сырья» щелкните по ней правой кнопкой мыши и выберите в контекстном меню **Cost**.

Откроется диалоговое окно для внесения стоимости затрат (рис. 98).

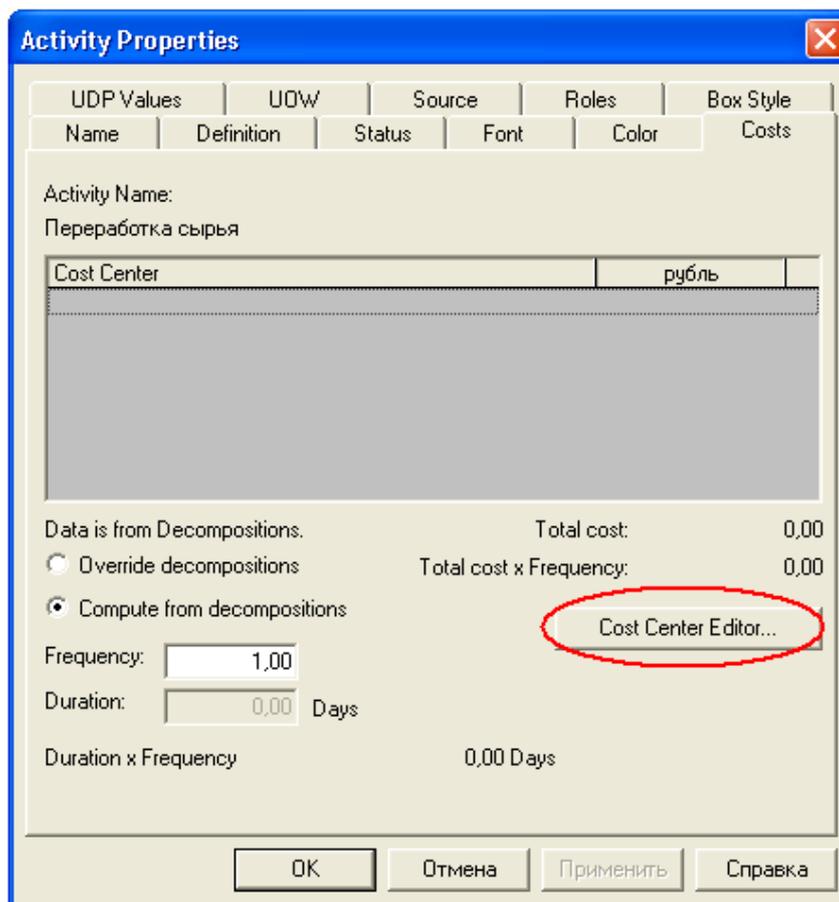


Рисунок 98 - Диалог Activity Properties

7. В диалоговом окне **Cost Center Editor** добавьте центры затрат «Компоненты», «Рабочая сила», «Управление» (рис. 99).

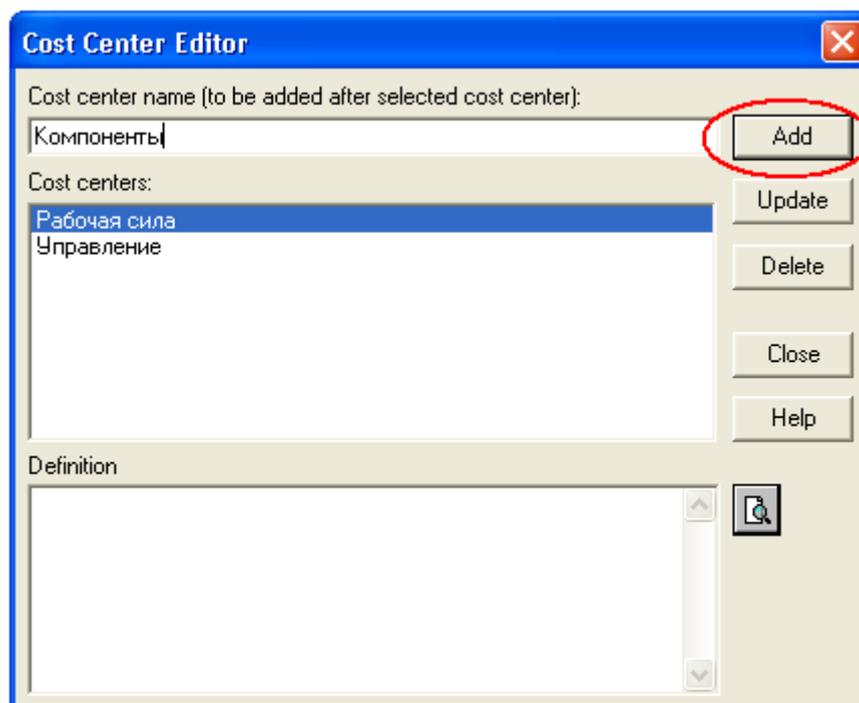


Рисунок 99 - Диалог Cost Center Editor

8. Во вкладке **Costs** диалога **Activity Properties** укажите частоту проведения данной работы в рамках общего процесса (**Frequency**) и продолжительность (**Duration**).

9. Выберите в списке один из центров затрат и в окне **Cost** задайте его стоимость (рис. 100).

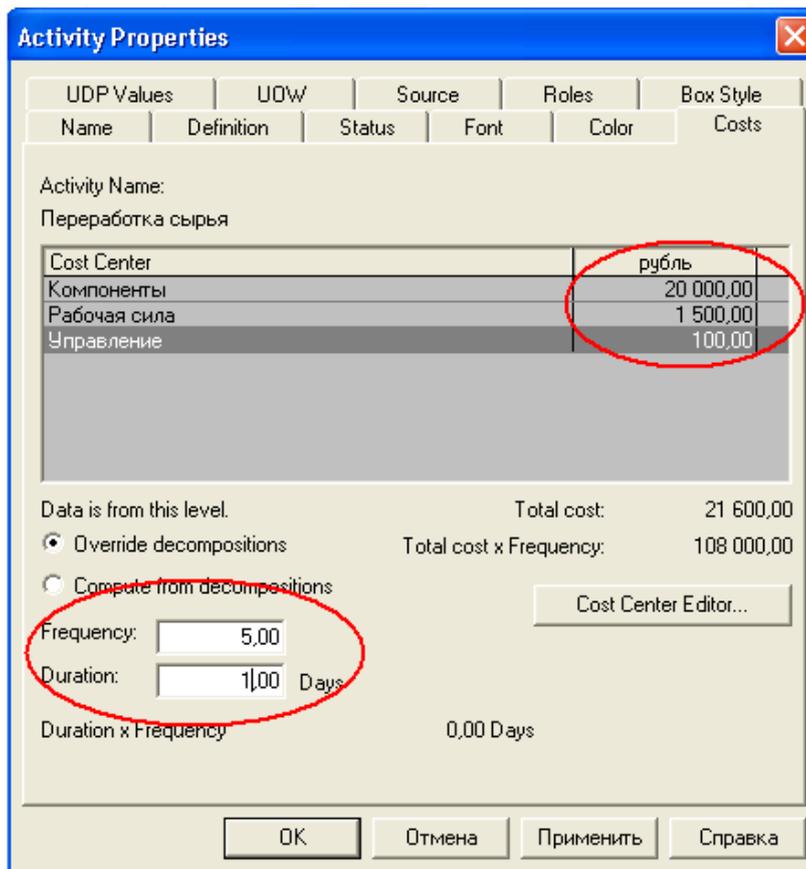


Рисунок 100 - Внесение стоимости в центры затрат



Задание 38. Внесение стоимости затрат.

1. Для работ «Изготовление деталей», «Сборка изделия», «Контроль качества» на диаграмме **A1** самостоятельно внесите параметры **ABC** из табл. 8.

Таблица 8 - Стоимости работ на диаграмме A1

Имя работы (Activity Name)	Центр затрат (Cost Center)	Сумма центра затрат (Cost Center Cost), руб.	Продолжитель- ность (Duration), день	Частота (Frequency)
Отслеживание расписания и	Управление	1500,00		

управление сборкой мебели и её проверкой				
Переработка сырья	Управление	100,00		
	Рабочая сила	1500,00	1,00	5,00
	Компоненты	20000,00		
Изготовление деталей	Управление	200,00		
	Рабочая сила	2500,00	2,00	5,00
	Компоненты	200,00		
Сборка мебели	Управление	600,00		
	Рабочая сила	1500,00	2,00	5,00
	Компоненты	200,00		
Контроль качества	Управление	600,00		
	Рабочая сила	1000,00	1,00	5,00
	Компоненты	0,00		

2. Проверьте себя (рис. 101).

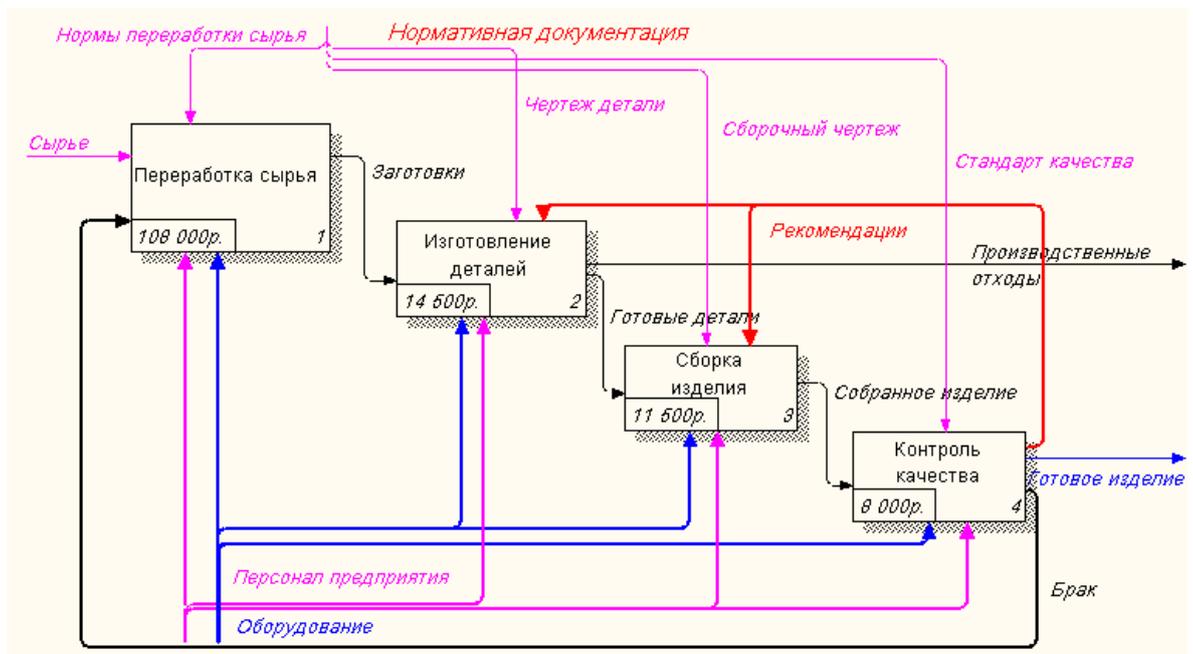


Рисунок 101 - Результат стоимости работ на диаграмме A1

3. Посмотрите результат – стоимость работы верхнего уровня (рис. 101).

 После того, как эта работа будет проделана, пригласите преподавателя и продемонстрируйте ему результат.



Задание 39. Составление отчета.

Для того чтобы сгенерировать отчет, выполните следующие действия:

1. Выберите пункт меню **Activity Cost Report (Tools – Reports - Activity Cost Report)**.
2. Укажите пункты в диалоге **Activity Based Costing Report** (рис. 102), по которым хотите получить сведения.

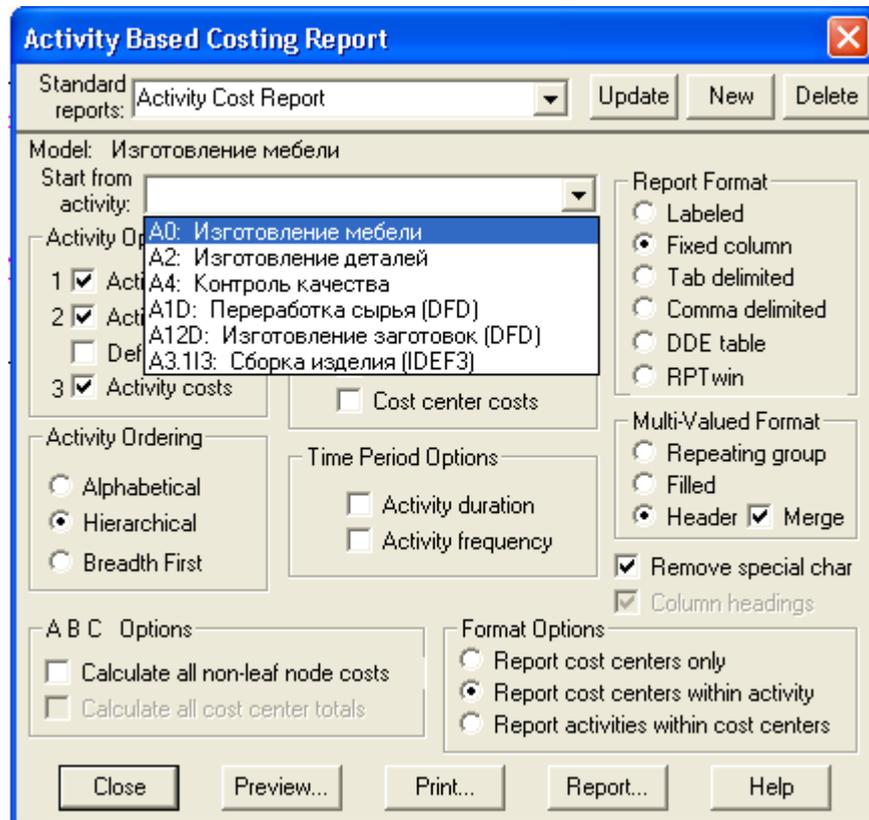


Рисунок 102 - Диалог Activity Based Costing Report

Назначение пунктов можно посмотреть в таблице 9.

Таблица 9 - Пункты Activity Based Costing Report и их значение

<p>Параметры работы</p> <p>Activity Options</p> <ul style="list-style-type: none"> <input checked="" type="checkbox"/> Activity name — Название работы <input checked="" type="checkbox"/> Activity number — Номер работы <input checked="" type="checkbox"/> Definition — Определение <input checked="" type="checkbox"/> Activity costs — Затраты на работу 	<p>Список работ</p> <p>Activity Ordering</p> <ul style="list-style-type: none"> <input type="checkbox"/> Alphabetical — По алфавиту <input checked="" type="checkbox"/> Hierarchical — Иерархический <input type="checkbox"/> Breadth First — Сначала (по порядку)
<p>Параметры центра стоимости</p> <p>Cost Center Options</p> <ul style="list-style-type: none"> <input type="checkbox"/> Cost center name — Имя центра стоимости <input type="checkbox"/> Cost center definition — Определение центра стоимости <input type="checkbox"/> Cost center costs — Центр стоимости затрат 	

Параметры времени

Time Period Options

Activity duration — Продолжительность работы

Activity frequency — Частота работы

Формат отчета

Report Format

Labeled — Отчеты включают метку поля, затем, в следующей строке, печатается содержимое поля

Fixed column — Каждое поле печатается в собственной колонке

Tab delimited — Каждое поле печатается в собственной колонке. Колонки разделяются знаком табуляции или запятыми

Comma delimited — Данные передаются по DDE приложению (в MS Word или Excel)

DDE table — Отчет создается в формате Platinum RPTwin - специализированного генератора отчетов BPwin

RPTwin

Параметры ABC

A B C Options

Calculate all non-leaf node costs — Вычисление затрат всех узлов

Calculate all cost center totals — Вычисление стоимости всех центров затрат

Вывод полей в отчете при группировке данных

Multi-Valued Format

Repeating group — Детальные данные объединяются в одно поле, между значениями вставляется +

Filled — Дублирование данных для каждого заголовка группы

Header Merge — Печатается заголовок группы, затем - детальная информация

Параметры формата

Format Options

Report cost centers only — Сообщение только центров стоимости

Report cost centers within activity — Сообщение центров стоимости о работе

Report activities within cost centers — Сообщение о работах, входящих в центры стоимости

3. Вот что должно у вас получиться (рис. 103).

Report Format: Column			
Activity Name	Activity Number	Activity Cost (рубль)	Cost Center
Изготовление мебели	0	142 000,00	Компоненты + Рабочая сила + Управление
Переработка сырья	1	21 600,00	Компоненты + Рабочая сила + Управление
Изготовление деталей	2	2 900,00	Компоненты + Рабочая сила + Управление
Сборка изделия	3	2 300,00	Компоненты + Рабочая сила + Управление
Контроль качества	4	1 600,00	Рабочая сила + Управление

Рисунок 103 - Сгенерированный отчет

 После того, как эта работа будет проделана, пригласите преподавателя и продемонстрируйте ему результат.



Задание 40. Сохранение полученной диаграммы.

Сохраните полученную диаграмму.

1. В меню **File** выберите **Save as**.
2. Укажите путь к своей папке и имя файла **Lab7.bp1**.
3. Нажмите ОК.

Контрольные вопросы

1. Дайте определение понятию стоимостный анализ.
2. Какова роль стоимостного анализа в моделировании бизнес-процессов?
3. Какие понятия включает в себя стоимостный анализ?
4. Какие действия в программе необходимо выполнить, чтобы получить отчет о стоимости бизнес-процесса?

Тестовое задание

1. На чем основан стоимостный анализ?

- a) на контекстной диаграмме;
- b) на модели работ;
- c) на хранилище данных;
- d) правильные ответы a), b);
- e) правильных ответов нет.

2. Укажите сокращенное название стоимостного анализа:

- a) DFD;
- b) ABC;
- c) IDEF0;
- d) IDEF3;
- e) UOW.

3. Какие понятия включает в себя стоимостный анализ?

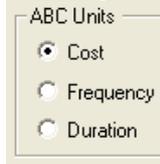
- a) объект затрат;
- b) движитель затрат;
- c) центры затрат;
- d) правильные ответы a), b);
- e) правильные ответы a), c).

4. Где на диаграмме отображается стоимость каждой работы?

- a) В центре работы;
- b) в левом нижнем углу работы;

- c) в левом верхнем углу работы;
- d) в правом верхнем углу работы;
- e) в правом нижнем углу работы.

5. Каково назначение радиокнопки *Frequency* в группе *ABC Units*



- a) обозначает стоимость работы;
- b) обозначает продолжительность работы;
- c) обозначает частоту работы;
- d) ничего не обозначает;
- e) правильного ответа нет.

6. Какую опцию на вкладке *Display (Model Properties)* надо включить, чтобы отображать стоимость работы на диаграмме?

- a) Colors;
- b) ABC Data;
- c) Arrow Names;
- d) Activity Numbers;
- e) Data Store Numbers.

7. В каком диалоговом окне можно настроить параметры отчета?

- a) Model Properties;
- b) Activity Based Costing Report;
- c) Activity Properties;
- d) Cost Center Editor;
- e) Cost Center Dictionary.

 После того, как эта работа будет проделана, пригласите преподавателя и продемонстрируйте ему результат.

Контрольное задание

Продолжите выполнение контрольного задания.

1. Откройте файл **Kr6.bp1**.
2. Произведите стоимостный анализ модели.
3. Сгенерируйте отчет в своем контрольном задании.
4. Сохраните файл в своей папке с именем **Kr7.bp1**.

 После того, как эта работа будет проделана, пригласите преподавателя и продемонстрируйте ему результат.

Литература

1. Вендров А.М. Case-технологии. Современные методы и средства проектирования информационных систем. - М.: Финансы и статистика, 1998. – 98 с.
2. Вендров А.М. Практикум по проектированию программного обеспечения экономических информационных систем: Учебное пособие. - М.: Финансы и статистика, 2006. – 192 с.
3. Вендров А.М. Проектирование программного обеспечения экономических информационных систем: Учебник - 2-е изд., перераб. и доп. - М.: Финансы и статистика, 2005. – 192 с.
4. Грекул В.И. Проектирование информационных систем. Интернет-университет информационных технологий / В.И. Грекул, Г.Н. Денищенко, Н.Л. Коровкина - ИНТУИТ.ру, 2005. – 296с. - www.intuit.ru.
5. Маклаков С.В. Моделирование бизнес-процессов с AIFusion Process Modeler. - М.: Диалог-МИФИ, 2004.
6. Маклаков С.В. BPwin и ERwin. CASE-средства разработки информационных систем. - 2-е изд., испр. и дополн. - М.: Диалог-МИФИ, 2001.
7. Федеральный закон Российской Федерации от 27 июля 2006 г. N 149-ФЗ. Об информации, информационных технологиях и о защите информации.
8. ГОСТ 24.202-80. Требования к содержанию документа «Технико-экономическое обоснование создания АСУ».
9. ГОСТ 34.201-89. Информационная технология. Комплекс стандартов на автоматизированные системы. Виды, комплектность и обозначение документов при создании автоматизированных систем.
10. ГОСТ 34.602-89. Информационная технология. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированных систем.

11. ГОСТ 34.003-90. Информационная технология. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Термины и определения.

12. ГОСТ 34.601-90. Информационная технология. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Стадия создания.

13. ГОСТ 34.603-92. Виды испытаний автоматизированных систем.

14. ГОСТ Р ИСО/МЭК 9126-93. Информационная технология. Оценка программной продукции. Характеристики качества и руководства по их применению.

15. ГОСТ Р ИСО/МЭК 12207-99 Информационная технология. Процессы жизненного цикла программных средств.

16. ГОСТ РВ 51987-2002. Информационная технология. Комплекс стандартов на автоматизированные системы. Требования и показатели качества функционирования информационных систем (ИС). Общие положения.

17. ГОСТ Р ИСО 15288-2005 Информационная технология. Системная инженерия. Процессы жизненного цикла систем.

18. Анисимов В.В. Проектирование информационных систем. Электронный ресурс: <https://sites.google.com/site/anisimovkhv/learning/pris>.

19. Галямина И.Г. Управление процессами - СПб.: Питер, 2013. - 304 с.

Надежда Владимировна Бендик
Асалханов Петр Георгиевич

Структурный подход к анализу и проектированию информационных систем
Учебно-методическое пособие

Лицензия на издательскую деятельность
ЛР № 070444 от 11.03.98 г.
Подписано в печать 10.04.2018 г.
Тираж 30 экз.

Издательство Иркутского государственного аграрного
университета имени А.А. Ежевского
664038, Иркутская обл., Иркутский р-н, пос. Молодежный