

Документ подписан проставив электронную подпись
Информация о владельце:
ФИО: Шамрай-Курбатова Лидия Викторовна
Должность: Ректор
Дата подписания: 09.06.2026 10:08:49
Уникальный программный ключ:
b1e4399771b07e18f31755456972d73b2ccfc531

Автономная некоммерческая организация высшего образования
«Волгоградский институт бизнеса»

Рабочая программа учебной дисциплины

Проектирование информационных систем

(Наименование дисциплины)

09.03.03 Прикладная информатика, направленность (профиль) «Прикладной искусственный интеллект»

(Направление подготовки / Профиль)

Бакалавр

(Квалификация)

Кафедра разработчик

Экономики и управления

Год набора

2026

Вид учебной деятельности	Трудоемкость (объем) дисциплины	
	Очная форма	Очно-заочная форма
	д	в
Зачетные единицы	3	3
Общее количество часов	108	108
Аудиторные часы контактной работы обучающегося с преподавателями:	32	16
– Лекционные (Л)	16	8
– Практические (ПЗ)	16	8
– Лабораторные (ЛЗ)		
– Семинарские (СЗ)		
Самостоятельная работа обучающихся (СРО)	22	56
К (Р-Г) Р (П) (+;-)		
Тестирование (+;-)		
ДКР (+;-)		
Зачет (+;-)		
Зачет с оценкой (+;- (Кол-во часов))		
Экзамен (+;- (Кол-во часов))	+ (54)	+ (36)

Волгоград 2026

Содержание

Раздел 1. Организационно-методический раздел	3
Раздел 2. Тематический план.....	5
Раздел 3. Содержание дисциплины.....	6
Раздел 4. Организация самостоятельной работы обучающихся.....	10
Раздел 5. Фонд оценочных средств для проведения текущего контроля и промежуточной аттестации обучающихся.....	12
Раздел 6. Оценочные средства промежуточной аттестации (с ключами)	19
Раздел 7. Перечень учебной литературы, необходимой для освоения дисциплины	47
Раздел 8. Материально-техническая база и информационные технологии.....	51
Раздел 9. Методические указания для обучающихся по освоению дисциплины	53

Раздел 1. Организационно-методический раздел

1.1. Цели освоения дисциплины

Дисциплина «Проектирование информационных систем» входит в обязательную **Б1.О.23** часть дисциплин, по направлению подготовки «**09.03.03 Прикладная информатика**», направленность (профиль) «**Прикладной искусственный интеллект**».

Целью дисциплины является формирование **компетенций** (в соответствии с ФГОС ВО и требованиями к результатам освоения основной профессиональной образовательной программы высшего образования (ОПОП ВО)):

ПК-3. Способен осуществлять проектирование компьютерного программного обеспечения;

Дескрипторы общепрофессиональной компетенции:

ПК-3.1. Способен проектировать архитектуру компьютерного программного обеспечения, включая интеллектуальные компоненты

ПК-3.2. Способен применять методы и средства проектирования программного обеспечения, включая проектирование интерфейсов и командную разработку

Перечисленные компетенции формируются в процессе достижения **индикаторов компетенций**:

Обобщенная трудовая функция/ трудовая функция	Код и наименование дескриптора компетенций	Код и наименование индикатора достижения компетенций (из ПС)
Тип задач проф. деятельности: проектный	<p>ПК-3.1. Способен проектировать архитектуру компьютерного программного обеспечения, включая интеллектуальные компоненты</p> <p>ПК-3.2. Способен применять методы и средства проектирования программного обеспечения, включая проектирование интерфейсов и командную разработку</p>	<p>Знает:</p> <p>ИД-1 ПК 3.1. Принципы построения и виды архитектуры компьютерного программного обеспечения</p> <p>ИД-2 ПК 3.2. Методы и средства проектирования компьютерного программного обеспечения</p> <p>Умеет:</p> <p>ИД-3 ПК 3.1. Использовать существующие типовые решения и шаблоны проектирования компьютерного программного обеспечения D/03.6</p> <p>ИД-4 ПК 3.2. Использовать командные средства разработки компьютерного программного обеспечения D/03.6</p> <p>Имеет навыки и (или) опыт:</p> <p>ИД-5 ПК 3.1. Разработки, изменения архитектуры компьютерного программного обеспечения и ее согласования с системным аналитиком и архитектором программного обеспечения D/03.6</p> <p>ИД-6 ПК 3.2. Проектирования программных интерфейсов D/03.6</p>

**1.2. Место дисциплины в структуре ОПОП ВО
направления подготовки «09.03.03 Прикладная информатика», направленность (профиль)
«Прикладной искусственный интеллект»**

№	Предшествующие дисциплины (дисциплины, изучаемые параллельно)	Последующие дисциплины
<i>1</i>	<i>2</i>	<i>3</i>
1	Введение в направление подготовки	ВКР
7	Иностранный язык	
8	Математика	
9	Информатика	
10	Вычислительные системы, сети и телекоммуникации	
11	Информационные технологии в менеджменте	
12	Базы данных	

Последовательность формирования компетенций в указанных дисциплинах может быть изменена в зависимости от формы и срока обучения, а также преподавания с использованием дистанционных технологий обучения.

1.3. Нормативная документация

Рабочая программа учебной дисциплины составлена на основе:

- Федерального государственного образовательного стандарта высшего образования по направлению подготовки **«09.03.03 Прикладная информатика»**;
- Учебного плана направления подготовки **«09.03.03 Прикладная информатика», направленность (профиль) «Прикладной искусственный интеллект»** 2026 года набора;
- Образца рабочей программы учебной дисциплины (утвержден приказом №113-О от 01.09.2021 г.).

Раздел 2. Тематический план

Очная форма обучения (полный срок)

№	Тема дисциплины	Трудоемкость				СРО	Код индикатора и дескриптора достижения компетенций
		Всего	Аудиторные занятия				
			Л	ПЗ (ЛЗ, СЗ)			
1	2	3	4	5	6	7	
1	Тема 1. Теоретические основы проектирования информационных систем	4	2		2	ИД-1 ПК- 3.1 ИД-2 ПК- 3.2	
2	Тема 2. Технологии проектирования ИС	4	2		2	ИД-3 ПК- 3.1 ИД-4 ПК- 3.2	
3	Тема 3. Стандарты и профили в области информационных систем	6	2	2	2	ИД-5 ПК- 3.1 ИД-6 ПК- 3.2	
4	Тема 4. Моделирование функциональной области внедрения ИС	8	2	4	2	ИД-1 ПК- 3.1 ИД-2 ПК- 3.2	
5	Тема 5. Каноническое проектирование ИС	6	2	2	2	ИД-3 ПК- 3.1 ИД-4 ПК- 3.2	
6	Тема 6 Автоматизированное проектирование ИС	6	2	2	2	ИД-5 ПК- 3.1 ИД-6 ПК- 3.2	
7	Тема 7. Типовое проектирование ИС	6	2	2	2	ИД-3 ПК- 3.1 ИД-4 ПК- 3.2	
8	Тема 8. Проектирование интегрированных информационных систем	8	2	2	4	ИД-5 ПК- 3.1 ИД-6 ПК- 3.2	
9	Тема 9. Управление проектированием ИС	6		2	4	ИД-5 ПК- 3.1 ИД-6 ПК- 3.2	
Вид промежуточной аттестации (Экзамен)		54					
Итого		108	16	16	22		

Очно-заочная форма обучения (полный срок)

№	Тема дисциплины	Трудоемкость				СРО	Код индикатора и дескриптора достижения компетенций
		Всего	Аудиторные занятия				
			Л	ПЗ (ЛЗ, СЗ)			
1	2	3	4	5	6	7	
1	Тема 1. Теоретические основы проектирования информационных систем	8	2		6	ИД-1 ПК- 3.1 ИД-2 ПК- 3.2	
2	Тема 2. Технологии проектирования ИС	8	2		6	ИД-3 ПК- 3.1 ИД-4 ПК- 3.2	
3	Тема 3. Стандарты и профили в области информационных систем	8		2	6	ИД-5 ПК- 3.1 ИД-6 ПК- 3.2	
4	Тема 4. Моделирование функциональной области внедрения ИС	8	2		6	ИД-1 ПК- 3.1 ИД-2 ПК- 3.2	

5	Тема 5. Каноническое проектирование ИС	7	1		6	ИД-3 ПК- 3.1 ИД-4 ПК- 3.2
6	Тема 6 Автоматизированное проектирование ИС	7	1		6	ИД-5 ПК- 3.1 ИД-6 ПК- 3.2
7	Тема 7. Типовое проектирование ИС	8		2	6	ИД-3 ПК- 3.1 ИД-4 ПК- 3.2
8	Тема 8. Проектирование интегрированных информационных систем	8		2	6	ИД-5 ПК- 3.1 ИД-6 ПК- 3.2
9	Тема 9. Управление проектированием ИС	10		2	8	ИД-5 ПК- 3.1 ИД-6 ПК- 3.2
Вид промежуточной аттестации (Экзамен)		36				
Итого		108	8	8	56	

Раздел 3. Содержание дисциплины

3.1. Содержание дисциплины

Тема 1. Теоретические основы проектирования информационных систем

Понятие ИС. Структура ИС. Основные понятия и структура проекта ИС. Требования к эффективности и надежности проектных решений. Жизненный цикл ИС. Модели Жизненного цикла. Принципы разработки презентаций ИС и начального обучения пользователей.

Тема 2. Технологии проектирования информационных систем

Основные компоненты технологии проектирования ИС. Методы и средства проектирования ИС. Характеристика применяемых технологий проектирования. Требования, предъявляемые к технологии проектирования ИС. Выбор технологии проектирования ИС.

Тема 3. Стандарты и профили в области информационных систем

Классификация стандартов на проектирование и разработку информационных систем. Международный стандарт ISO/IEC 12207: 1995-08-01 Стандарты комплекса ГОСТ34 Методика Oracle CDM. Rational Unified Process (RUP). Microsoft Solution Framework (MSF). Extreme Programming (XP). Понятие профиля ИС. Процессы формирования, развития и применения профилей информационных систем.

Тема 4. Моделирование функциональной области внедрения информационных систем

Анализ и моделирование функциональной области внедрения информационных систем. Методологии моделирования предметной области. Классификация методологий. Сравнительный анализ технологий. Методология функционального моделирования. Методология объектно - ориентированного моделирования.

Тема 5. Каноническое проектирование информационных систем

Стадии и этапы процесса проектирования ИС. Состав работ на предпроектной стадии, стадии технического и рабочего проектирования, стадии ввода в действие ИС, эксплуатации и сопровождения. Состав проектной документации. Состав, содержание и принципы организации информационного обеспечения ИС. Проектирование пользовательского интерфейса. Проектирование документальных и фактографических БД.

Тема 6. Автоматизированное проектирование информационных систем

Основные принципы Case-технологии. Факторы эффективности Case-технологии. Классификация Case-средств проектирования и стратегия их выбора. Функционально-ориентированный подход. Этапы проектирования. Объектно-ориентированный подход. Содержание RAD-технологии прототипного создания приложений.

Тема 7. Типовое проектирование информационных систем

Понятие типового элемента. Классификация и примеры типовых информационных систем и их характеристика. Методы конфигурирования типовой информационной системы. Технологии параметрически - ориентированного и модельно-ориентированного проектирования.

Тема 8. Проектирование интегрированных информационных систем

Принципы и особенности проектирования интегрированных ИС. Открытые информационные системы: основные свойства и межсистемные интерфейсы Стандартные методы совместного доступа к базам и программам в сложных информационных системах. Система управления информационными потоками как средство интеграции приложений ИС. Методы и средства организации метаинформации проекта ИС. Проектирование системы защиты данных в ИС.

3.2. Содержание практического блока дисциплины

Очная форма обучения (полный срок)

№	Тема практического (семинарского, лабораторного) занятия
1	2
Тема 3. Стандарты и профили в области информационных систем	
ПЗ 1	Методика Oracle CDM
Тема 4. Моделирование функциональной области внедрения ИС	
ПЗ 2	Концептуальное моделирование предметной области
ПЗ 3	Основы работы в СА ERwin Process Modeler (Prwin). – инструмент реализации методологий структурного анализа и проектирования
ПЗ 4	Функциональное моделирование (IDEF0). Синтаксис и семантика моделей IDEF0
Тема 5. Каноническое проектирование ИС	
ПЗ 5	Состав проектной документации. Формирование отчета об обследовании
Тема 6. Автоматизированное проектирование ИС	
ПЗ 6	Основы работы в СА ERwin Data Modeler (ERWIN). Уровни проектирования баз данных
Тема 7. Типовое проектирование ИС	
ПЗ 7	Работа с типовыми проектными решениями
Тема 8. Проектирование интегрированных информационных систем	
ПЗ 8	Документирование модели данных в СА ERwin Data Modeler (ERWIN)
Тема 9. Управление проектированием ИС	
ПЗ 9	Управление проектом. Инструменты. Методы

Очно-заочная форма обучения (полный срок)

№	Тема практического (семинарского, лабораторного) занятия
1	2
Тема 3. Стандарты и профили в области информационных систем	
ПЗ 1	Методика Oracle CDM
Тема 7. Типовое проектирование ИС	
ПЗ 7	Работа с типовыми проектными решениями
Тема 8. Проектирование интегрированных информационных систем	
ПЗ 8	Документирование модели данных в СА ERwin Data Modeler (ERWIN)
Тема 9. Управление проектированием ИС	
ПЗ 9	Управление проектом. Инструменты. Методы

3.3. Образовательные технологии

Очная форма обучения (полный срок)

№	Тема занятия	Вид учебного занятия	Форма / Методы интерактивного обучения	% учебного времени
1	2	3	4	5
1	Тема 1. Теоретические основы проектирования информационных	ПЗ	Метод проектов	75

	систем			
2	Тема 2. Технологии проектирования ИС	ПЗ	Метод проектов	
3	Тема 3. Стандарты и профили в области информационных систем	ПЗ	Метод проектов	75
4	Тема 4. Моделирование функциональной области внедрения ИС	ПЗ	Метод проектов	75
5	Тема 5. Каноническое проектирование ИС	ПЗ	Метод проектов	75
6	Тема 6 Автоматизированное проектирование ИС	ПЗ	Метод проектов	75
7	Тема 7. Типовое проектирование ИС	ПЗ	Метод проектов	75
8	Тема 8. Проектирование интегрированных информационных систем	ПЗ	Метод проектов	75
Итого %				25%

Очно-заочная форма обучения (полный срок)

№	Тема занятия	Вид учебного занятия	Форма / Методы интерактивного обучения	% учебного времени
<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>
1	Тема 1. Теоретические основы проектирования информационных систем	ПЗ	Метод проектов	75
2	Тема 2. Технологии проектирования ИС	ПЗ	Метод проектов	
4	Тема 4. Моделирование функциональной области внедрения ИС	ПЗ	Метод проектов	75
5	Тема 5. Каноническое проектирование ИС	ПЗ	Метод проектов	75
6	Тема 6 Автоматизированное проектирование ИС	ПЗ	Метод проектов	75
Итого %				25%

Раздел 4. Организация самостоятельной работы обучающихся

4.1. Организация самостоятельной работы обучающихся

№	Тема дисциплины	№ вопросов	№ рекомендуемой литературы
1	2	3	4
1	Тема 1. Теоретические основы проектирования информационных систем	1, 26, 27	1, 2, 5, 6
2	Тема 2. Технологии проектирования ИС	2,3	1, 2, 3, 5
3	Тема 3. Стандарты и профили в области информационных систем	4,5	1, 2, 3, 5
4	Тема 4. Моделирование функциональной области внедрения ИС	6,7,8,9,10	2, 3, 4, 7
5	Тема 5. Каноническое проектирование ИС	11,12,13,14,15	1, 2, 3, 5
6	Тема 6 Автоматизированное проектирование ИС	16,17,18	1, 2, 3, 5
7	Тема 7. Типовое проектирование ИС	19	1, 2, 3, 5
8	Тема 8. Проектирование интегрированных информационных систем	20,21	1, 2, 3, 5
9	Тема 9. Управление проектированием ИС	22,23	1, 2, 3, 5

Перечень вопросов, выносимых на самостоятельную работу обучающихся

1. Основные принципы системного подхода к созданию ИС.
2. Классификация технологий проектирования ИС.
3. Классификация средств проектирования ИС.
4. Классификация стандартов на проектирование и разработку информационных систем.
5. Процессы формирования, развития и применения профилей информационных систем.
6. Классификация структурных методологий. Сравнительный анализ структурных методологий.
7. Методологии структурного анализа Йодана/де Марко и Гейна-Сарсона (DFD - технология).
8. Диаграммы потоков данных: словари данных, спецификации процессов.
9. Диаграммы переходов состояний. Назначение, объекты, правила и способы построения.
10. Методология объектно - ориентированного моделирования.
11. Проектирование документальных и фактографических БД.
12. Назначение технико-экономического обоснования, его основные компоненты.
13. Назначение технического задания.
14. Методы внедрения проекта ЭИС и их особенности.
15. Внемашиное информационное обеспечение ИС, его компоненты.
16. Факторы эффективности Case-технологии.
17. Классификация CASE-средств, стратегия их выбора.
18. Объектно-ориентированная технология проектирования в CASE-системах.
19. Классификация и примеры типовых информационных систем и их характеристика.
20. Принципы проектирования интегрированных ИС.
21. Система управления информационными потоками как средство интеграции приложений ИС.
22. Требования к эффективности и надежности проектных решений.
23. Планирование и контроль проектных работ.
24. Показатели экономической эффективности информационной системы.
25. Оценка научно-технического уровня информационных систем.
26. Принципы построения презентаций ИС
27. Методика начального обучения пользователей

4.2. Перечень учебно-методического обеспечения самостоятельной работы обучающихся

Самостоятельная работа обучающихся обеспечивается следующими учебно-методическими материалами:

1. Указаниями в рабочей программе по дисциплине (п.4.1.)

2. Лекционные материалы в составе учебно-методического комплекса по дисциплине
3. Заданиями и методическими рекомендациями по организации самостоятельной работы обучающихся в составе учебно-методического комплекса по дисциплине
4. Глоссарием по дисциплине в составе учебно-методического комплекса по дисциплине
5. Методическими указаниями по выполнению курсовой работы по дисциплине «Проектирование информационных систем»

Раздел 5. Фонд оценочных средств для проведения текущего контроля и промежуточной аттестации обучающихся

Фонд оценочных средств по дисциплине представляет собой совокупность контролирующих материалов, предназначенных для измерения уровня достижения обучающимися установленных результатов образовательной программы. ФОС по дисциплине используется при проведении оперативного контроля и промежуточной аттестации обучающихся. Требования к структуре и содержанию ФОС дисциплины регламентируются Положением о фонде оценочных материалов по программам высшего образования – программам бакалавриата, магистратуры.

5.1. Паспорт фонда оценочных средств

Очная форма обучения (полный срок)

№	Контролируемые разделы (темы) дисциплины	Оценочные средства			
		Л	ПЗ (ЛЗ, СЗ)	СРО	Код индикатора и дескриптора достижения компетенций
1	2	3	4	5	6
1	Теоретические основы проектирования информационных систем	УО		ПРВ	ИД-1 ПК- 3.1 ИД-2 ПК- 3.2
2	Технологии проектирования ИС	УО		ПРВ	ИД-3 ПК- 3.1 ИД-4 ПК- 3.2
3	Стандарты и профили в области информационных систем	УО	МШ	ПРВ	ИД-5 ПК- 3.1 ИД-6 ПК- 3.2
4	Моделирование функциональной области внедрения ИС	УО	МП	ПРВ	ИД-1 ПК- 3.1 ИД-2 ПК- 3.2
5	Каноническое проектирование ИС	УО	МП	ПРВ	ИД-3 ПК- 3.1 ИД-4 ПК- 3.2
6	Автоматизированное проектирование ИС	УО	МП	ПРВ	ИД-5 ПК- 3.1 ИД-6 ПК- 3.2
7	Типовое проектирование ИС	УО	УО	ПРВ	ИД-3 ПК- 3.1 ИД-4 ПК- 3.2
8	Проектирование интегрированных информационных систем	УО	МП	ПРВ	ИД-5 ПК- 3.1 ИД-6 ПК- 3.2
9	Управление проектированием ИС	УО	МП	ПРВ	ИД-5 ПК- 3.1 ИД-6 ПК- 3.2

Очно-заочная форма обучения (полный срок)

№	Контролируемые разделы (темы) дисциплины	Оценочные средства			
		Л	ПЗ (ЛЗ, СЗ)	СРО	Код индикатора и дескриптора достижения компетенций
1	2	3	4	5	6
1	Теоретические основы проектирования информационных систем	УО		ПРВ	ИД-1 ПК- 3.1 ИД-2 ПК- 3.2
2	Технологии проектирования ИС	УО		ПРВ	ИД-3 ПК- 3.1 ИД-4 ПК- 3.2
3	Стандарты и профили в области		МШ	ПРВ	ИД-5 ПК- 3.1

	информационных систем				ИД-6 ПК- 3.2
4	Моделирование функциональной области внедрения ИС	УО		ПРВ	ИД-1 ПК- 3.1 ИД-2 ПК- 3.2
5	Каноническое проектирование ИС	УО		ПРВ	ИД-3 ПК- 3.1 ИД-4 ПК- 3.2
6	Автоматизированное проектирование ИС	УО		ПРВ	ИД-5 ПК- 3.1 ИД-6 ПК- 3.2
7	Типовое проектирование ИС		УО	ПРВ	ИД-3 ПК- 3.1 ИД-4 ПК- 3.2
8	Проектирование интегрированных информационных систем		МП	ПРВ	ИД-5 ПК- 3.1 ИД-6 ПК- 3.2
9	Управление проектированием ИС		МП	ПРВ	ИД-5 ПК- 3.1 ИД-6 ПК- 3.2

Условные обозначения оценочных средств (Столбцы 3, 4, 5):

УО – Устный (фронтальный, индивидуальный, комбинированный) опрос

ПРВ – Проверка рефератов, отчетов, рецензий, аннотаций, конспектов, графического материала, эссе, переводов, решений заданий, выполненных заданий в электронном виде и т.д.

МШ – Метод мозгового штурма

МП – Метод проектов

5.2. Оценочные средства текущего контроля

Перечень практических (семинарских) заданий

Тема № 4: «Моделирование функциональной области внедрения ИС»

Установите триал версию программы AllFusion ERWin Process Modeler или BРwin.

Выполните моделирование согласно приведенному описанию.

BРwin имеет достаточно простой и интуитивно понятный интерфейс пользователя. При запуске BРwin по умолчанию появляется основная панель инструментов, палитра инструментов (вид которой зависит от выбранной нотации) и, в левой части, навигатор модели - Model Explorer (рис.1).

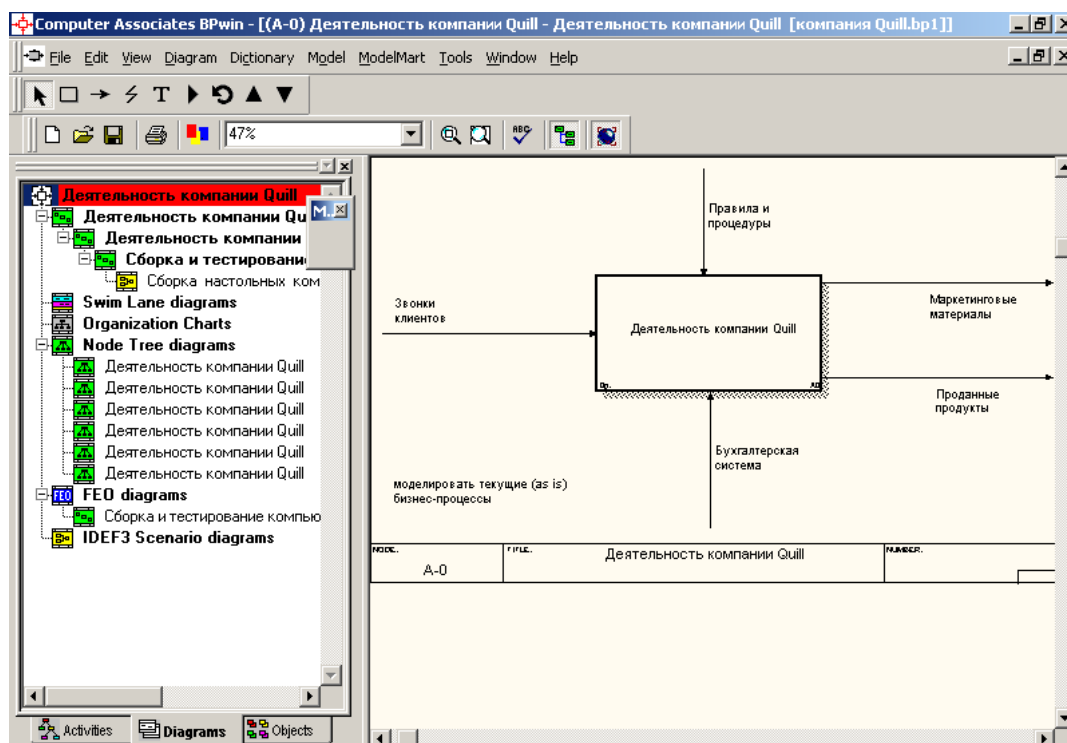


Рисунок 1. Интегрированная среда разработки модели Bрwin

При создании новой модели возникает диалог, в котором следует указать, будет ли создана модель заново или она будет открыта из файла либо из репозитория Model Mart, затем внести имя модели и выбрать методологию, в которой будет построена модель (рис. 2).

Как было указано выше, BPwin поддерживает три методологии - IDEF0, IDEF3 и DFD, каждая из которых решает свои специфические задачи. В BPwin возможно построение смешанных моделей, т. е. модель может содержать одновременно диаграммы как IDEF0, так и IDEF3 и DFD. Состав палитры инструментов изменяется автоматически, когда происходит переключение с одной нотации на другую.

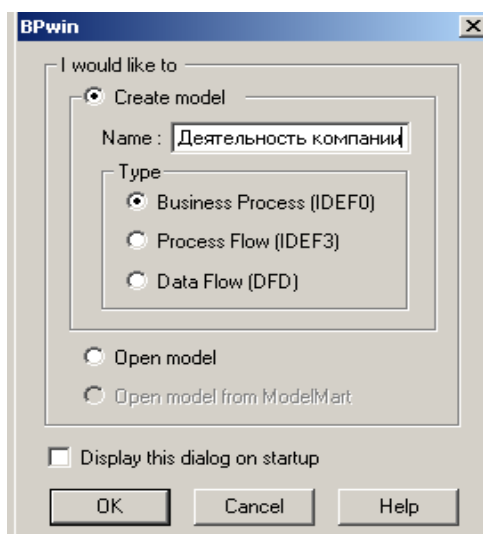


Рисунок 2. Диалог создания модели

Модель в BPwin рассматривается как совокупность работ, каждая из которых оперирует с некоторым набором данных. Работа изображается в виде прямоугольников, данные - в виде стрелок. Если щелкнуть по любому объекту модели левой кнопкой мыши, появляется контекстное меню, каждый пункт которого соответствует редактору какого-либо свойства объекта.

Построение модели IDEF0

На начальных этапах создания ИС необходимо понять, как работает организация, которую собираются автоматизировать. Руководитель хорошо знает работу в целом, но не в состоянии вникнуть в детали работы каждого рядового сотрудника. Рядовой сотрудник хорошо знает, что творится на его рабочем месте, но может не знать, как работают коллеги. Поэтому для описания работы предприятия необходимо построить модель, которая будет адекватна предметной области и содержать в себе знания всех участников бизнес-процессов организации.

Наиболее удобным языком моделирования бизнес-процессов является IDEF0, где система представляется как совокупность взаимодействующих работ или функций. Такая чисто функциональная ориентация является принципиальной - функции системы анализируются независимо от объектов, которыми они оперируют. Это позволяет более четко смоделировать логику и взаимодействие процессов организации.

Процесс моделирования системы в IDEF0 начинается с создания **контекстной диаграммы - диаграммы наиболее абстрактного уровня описания системы в целом, содержащей определение субъекта моделирования, цели и точки зрения на модель.**

Под субъектом понимается сама система, при этом необходимо точно установить, что входит в систему, а что лежит за ее пределами, другими словами, определить, что будет в дальнейшем рассматриваться как компоненты системы, а что как внешнее воздействие. На определение субъекта системы будут существенно влиять позиция, с которой рассматривается система, и цель моделирования - вопросы, на которые построенная модель должна дать ответ. Другими словами, вначале необходимо определить область моделирования. Описание области как системы в целом, так и ее компонентов является основой построения модели. Хотя предполагается, что в ходе моделирования область может корректироваться, она должна быть в основном сформулирована изначально, поскольку именно область определяет направление моделирования. При формулировании области необходимо учитывать два компонента - широту и глубину. Широта подразумевает определение границ модели - что будет рассматриваться внутри

системы, а что снаружи. Глубина определяет, на каком уровне детализации модель является завершенной. При определении глубины системы необходимо помнить об ограничениях времени - трудоемкость построения модели растет в геометрической прогрессии с увеличением глубины декомпозиции. После определения границ модели предполагается, что новые объекты не должны вноситься в моделируемую систему.

Цель моделирования. Цель моделирования определяется из ответов на следующие вопросы:

- Почему этот процесс должен быть смоделирован?
- Что должна показывать модель?
- Что может получить клиент?

Точка зрения (Viewpoint). Под точкой зрения понимается перспектива, с которой наблюдалась система при построении модели. Хотя при построении модели учитываются мнения различных людей, все они должны придерживаться единой точки зрения на модель. Точка зрения должна соответствовать цели и границам моделирования. Как правило, выбирается точка зрения человека, ответственного за моделируемую работу в целом.

IDEF0-модель предполагает наличие четко сформулированной цели, единственного субъекта моделирования и одной точки зрения. Для внесения области, цели и точки зрения в модели IDEF0 в BPwin следует выбрать пункт меню Model/Model Properties, вызывающий диалог Model Properties (рис. 3). В закладке Purpose следует внести цель и точку зрения, а в закладку Definition - определение модели и описание области.

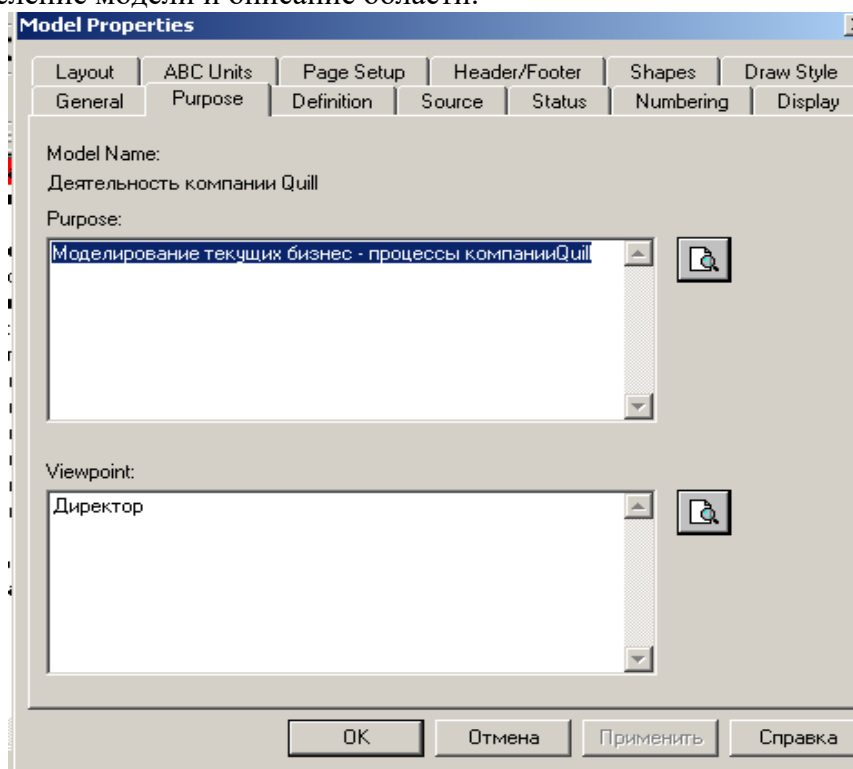


Рисунок 3. Диалог задания свойств модели

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

Модели AS-IS и TO-BE. Обычно сначала строится модель существующей организации работы - AS-IS (как есть). Анализ функциональной модели позволяет понять, где находятся наиболее слабые места, в чем будут состоять преимущества новых бизнес-процессов и насколько глубоким изменениям подвергнется существующая структура организации бизнеса. Детализация бизнес-процессов позволяет выявить недостатки организации даже там, где функциональность на первый взгляд кажется очевидной. Найденные в модели AS-IS недостатки можно исправить при создании модели TO-BE (как будет) - модели новой организации бизнес-процессов.

Технология проектирования ИС подразумевает сначала создание модели AS-IS, ее анализ и улучшение бизнес-процессов, то есть создание модели TO-BE, и только на основе модели TO-BE строится модель данных, прототип и затем окончательный вариант ИС.

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

Результат описания модели можно получить в отчете Model Report. Диалог настройки отчета по модели вызывается из пункта меню Tools/Reports/Model Report .

В диалоге настройки следует выбрать необходимые поля, при этом автоматически отображается очередность вывода информации в отчет (рис. 4).

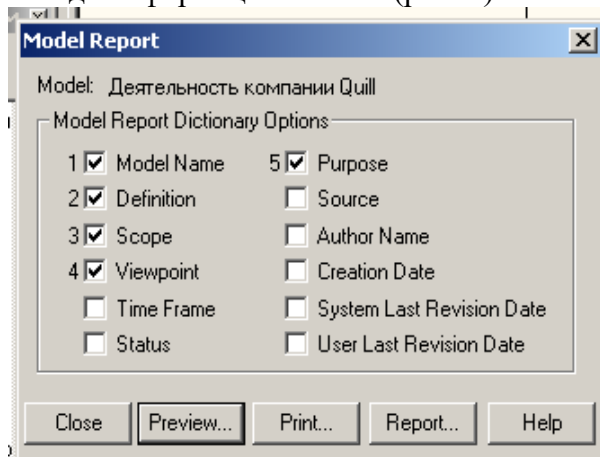


Рисунок 4. Диалоговое окно для формирования отчета по модели

На рис. 5 представлен отчет, сформированный по вышеуказанным полям.

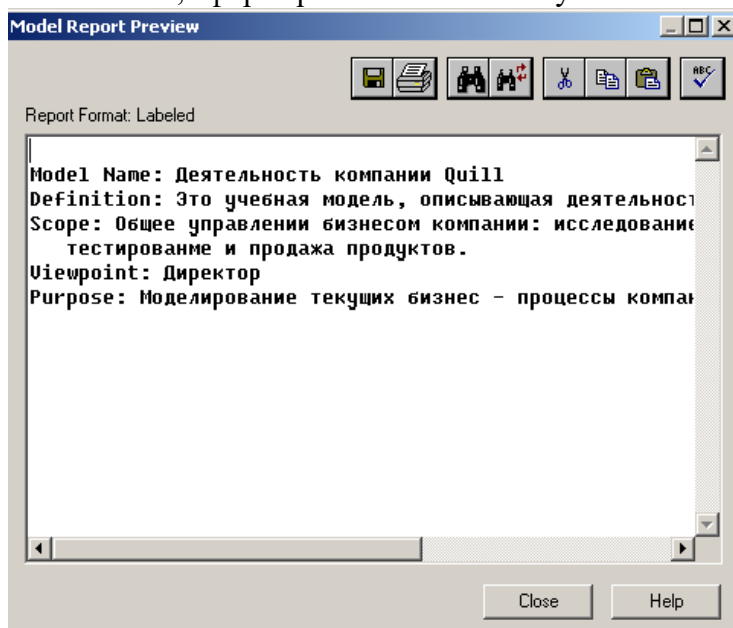


Рисунок 5. Предварительный просмотр отчета

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

Модель может содержать четыре типа диаграмм:

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

- диаграммы дерева узлов;
- диаграммы только для экспозиции (FEO).

Контекстная диаграмма является вершиной древовидной структуры диаграмм и представляет собой самое общее описание системы и ее взаимодействия с внешней средой. После описания системы в целом проводится разбиение ее на крупные фрагменты. Этот процесс называется функциональной декомпозицией, а диаграммы, которые описывают каждый фрагмент и взаимодействие фрагментов, называются диаграммами декомпозиции. После декомпозиции контекстной диаграммы проводится декомпозиция каждого большого фрагмента системы на более мелкие и так далее, до достижения нужного уровня подробности описания. После каждого сеанса декомпозиции проводятся сеансы экспертизы - эксперты предметной области указывают на соответствие реальных бизнес-процессов созданным диаграммам. Найденные несоответствия исправляются, и только после прохождения экспертизы без замечаний можно приступить к следующему сеансу декомпозиции. Так достигается соответствие модели реальным бизнес-процессам на любом и каждом уровне модели. Синтаксис описания системы в целом и каждого ее фрагмента одинаков во всей модели.

Диаграмма дерева узлов показывает иерархическую зависимость работ, но не взаимосвязи между работами. Диаграмм деревьев узлов может быть в модели сколь угодно много, поскольку дерево может быть построено на произвольную глубину и не обязательно с корня.

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

Работы (Activity). Работы обозначают поименованные процессы, функции или задачи, которые происходят в течение определенного времени и имеют распознаваемые результаты. Работы изображаются в виде прямоугольников. Все работы должны быть названы и определены. Имя работы должно быть выражено отглагольным существительным, обозначающим действие (например, "Деятельность компании", "Прием заказа" и т.д.). Работа "Деятельность компании" может иметь, например, следующее определение: "Это учебная модель, описывающая деятельность компании". При создании новой модели (меню File/New) автоматически создается контекстная диаграмма с единственной работой, изображающей систему в целом (рис. 6).

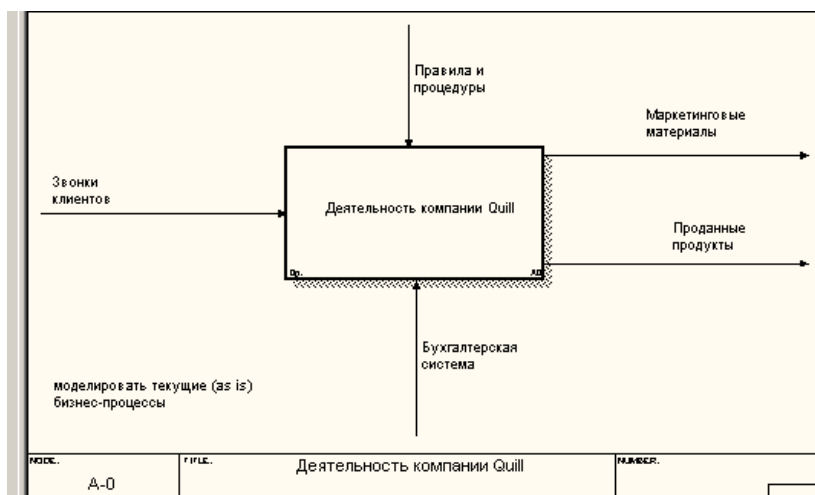


Рисунок 6. Пример контекстной диаграммы

Для внесения имени работы следует щелкнуть по работе правой кнопкой мыши, выбрать в меню Name Editor и в появившемся диалоге внести имя работы. Для описания других свойств работы служит диалог Activity Properties (рис. 7).

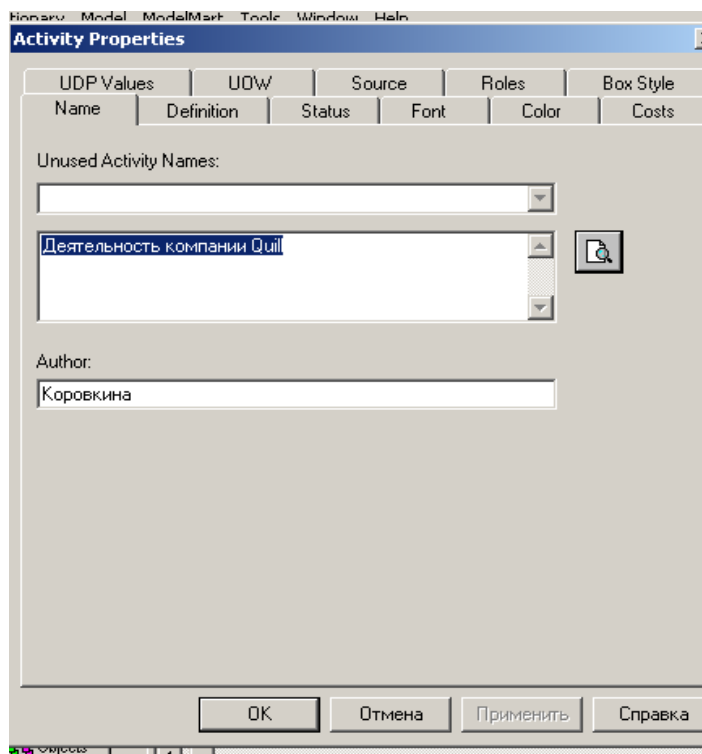



Рисунок Редактор задания свойств работы

Диаграммы декомпозиции содержат родственные работы, т.е. дочерние работы, имеющие общую родительскую работу. Для создания диаграммы декомпозиции следует щелкнуть по

кнопке  на панели инструментов

Возникает диалог Activity Box Count (рис. 8), в котором следует указать нотацию новой диаграммы и количество работ на ней. Остановимся пока на нотации IDEF0 и щелкнем на ОК. Появляется диаграмма декомпозиции (рис. 9). Допустимый интервал числа работ 2-8. Декомпозировать работу на одну работу не имеет смысла: диаграммы с количеством работ более восьми получаются перенасыщенными и плохо читаются. Для обеспечения наглядности и лучшего понимания моделируемых процессов рекомендуется использовать от трех до шести блоков на одной диаграмме.



Рисунок 8. Диалог Activity Box Count

Если оказывается, что количество работ недостаточно, то работу можно добавить в диаграмму, щелкнув сначала по кнопке

на палитре инструментов, а затем по свободному месту на диаграмме.

Работы на диаграммах декомпозиции обычно располагаются по диагонали от левого верхнего угла к правому нижнему.

Такой порядок называется порядком доминирования. Согласно этому принципу расположения в левом верхнем углу помещается самая важная работа или работа, выполняемая по времени первой. Далее вправо вниз располагаются менее важные или выполняемые позже работы. Такое размещение облегчает чтение диаграмм, кроме того, на нем основывается понятие взаимосвязей работ (см. ниже).

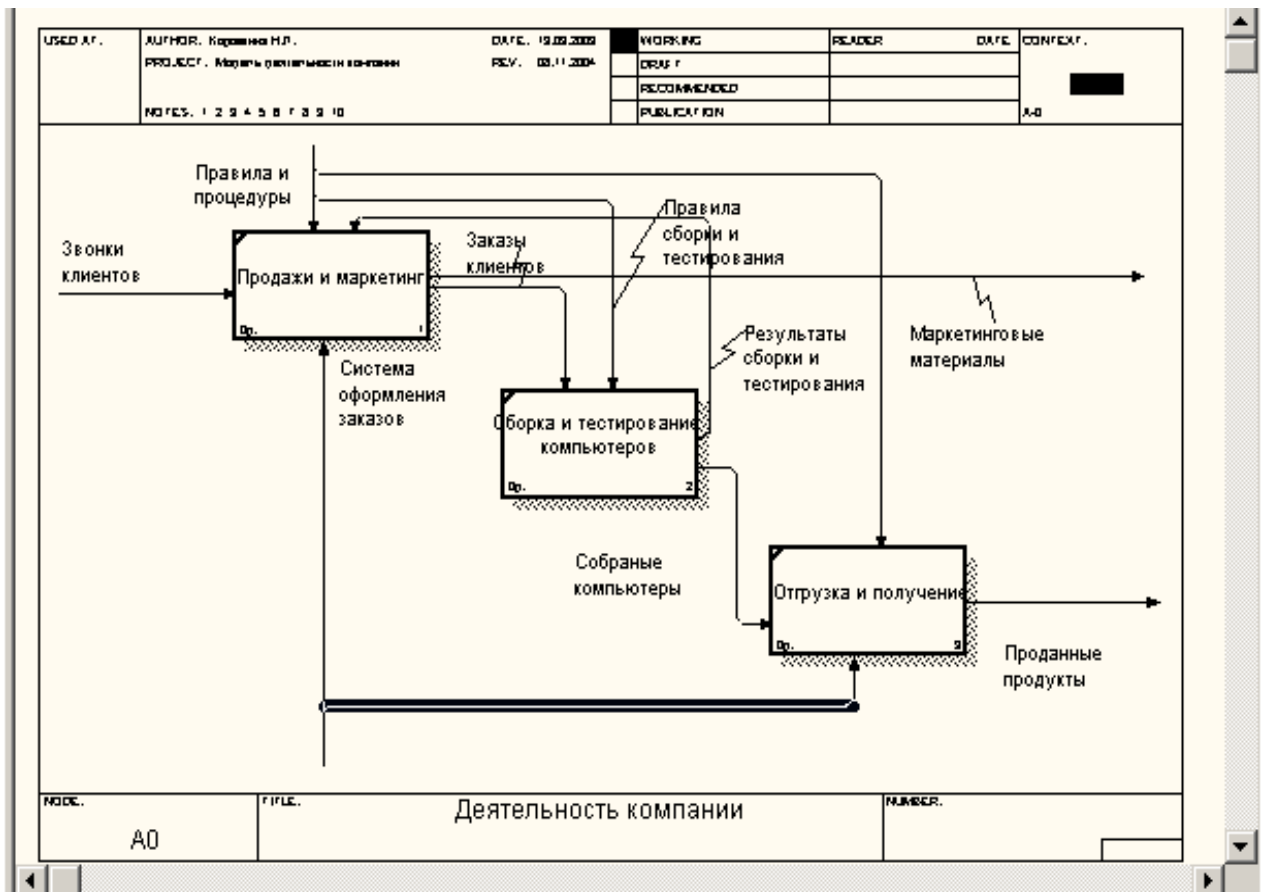


Рисунок 9. Пример диаграммы декомпозиции

Каждая из работ на диаграмме декомпозиции может быть в свою очередь декомпозирована. На диаграмме декомпозиции работы нумеруются автоматически слева направо. Номер работы показывается в правом нижнем углу. В левом верхнем углу изображается небольшая диагональная черта, которая показывает, что данная работа не была декомпозирована. Так, на рис. 9 все работы еще не были декомпозированы.

Стрелки (Arrow). Стрелки *описывают взаимодействие работ и представляют собой некую информацию, именуемую существительными.* (Например, "Звонки клиентов", "Правила и процедуры", "Бухгалтерская система".)

В IDEF0 различают пять типов стрелок:

Вход (Input) - материал или информация, которые используются или преобразуются работой для получения результата (выхода). Допускается, что работа может не иметь ни одной стрелки входа. Каждый тип стрелок подходит к определенной стороне прямоугольника, изображающего работу, или выходит из нее. Стрелка входа рисуется как входящая в левую грань работы. При описании технологических процессов (для этого и был придуман IDEF0) не возникает проблем определения входов. Действительно, "Звонки клиентов" на рис. 6 - это нечто, что перерабатывается в процессе "Деятельность компании" для получения результата. При моделировании ИС, когда стрелками являются не физические объекты, а данные, не все так очевидно. Например, при "Приеме пациента" карта пациента может быть и на входе и на выходе, между тем качество этих данных меняется. Другими словами, в нашем примере для того, чтобы оправдать свое назначение, стрелки входа и выхода должны быть точно определены с тем, чтобы указать на то, что данные действительно были переработаны (например, на выходе - "Заполненная карта пациента"). Очень часто сложно определить, являются ли данные входом или управлением. В этом случае подсказкой может служить, перерабатываются/изменяются ли данные в работе или нет. Если изменяются, то, скорее всего, это вход, если нет - управление.

Управление (Control) - правила, стратегии, процедуры или стандарты, которыми руководствуется работа. Каждая работа должна иметь хотя бы одну стрелку управления. Стрелка управления рисуется как входящая в верхнюю грань работы. На рис. 6 стрелка "Правила и процедуры" - управление для работы "Деятельность компании". Управление влияет на работу, но

не преобразуется работой. Если цель работы - изменить процедуру или стратегию, то такая процедура или стратегия будет для работы входом. В случае возникновения неопределенности в статусе стрелки (управление или вход) рекомендуется рисовать стрелку управления.

Выход (Output) - материал или информация, которые производятся работой. Каждая работа должна иметь хотя бы одну стрелку выхода. Работа без результата не имеет смысла и не должна моделироваться. Стрелка выхода рисуется как исходящая из правой грани работы. На рис. 6 стрелки "Маркетинговые материалы" и "Проданные продукты" являются выходом для работы "Деятельность компании".

Механизм (Mechanism) - ресурсы, которые выполняют работу, например персонал предприятия, станки, устройства и т. д. Стрелка механизма рисуется как входящая в нижнюю грань работы. На рис. 6 стрелка "Бухгалтерская система" является механизмом для работы "Деятельность компании". По усмотрению аналитика стрелки механизма могут не изображаться в модели.

Вызов (Call) - специальная стрелка, указывающая на другую модель работы. Стрелка вызова рисуется как исходящая из нижней грани работы. На рис. 10 стрелка "Другая модель работы" является вызовом для работы "Изготовление изделия". Стрелка вызова используется для указания того, что некоторая работа выполняется за пределами моделируемой системы. В VPwin стрелки вызова используются в механизме слияния и разделения моделей.

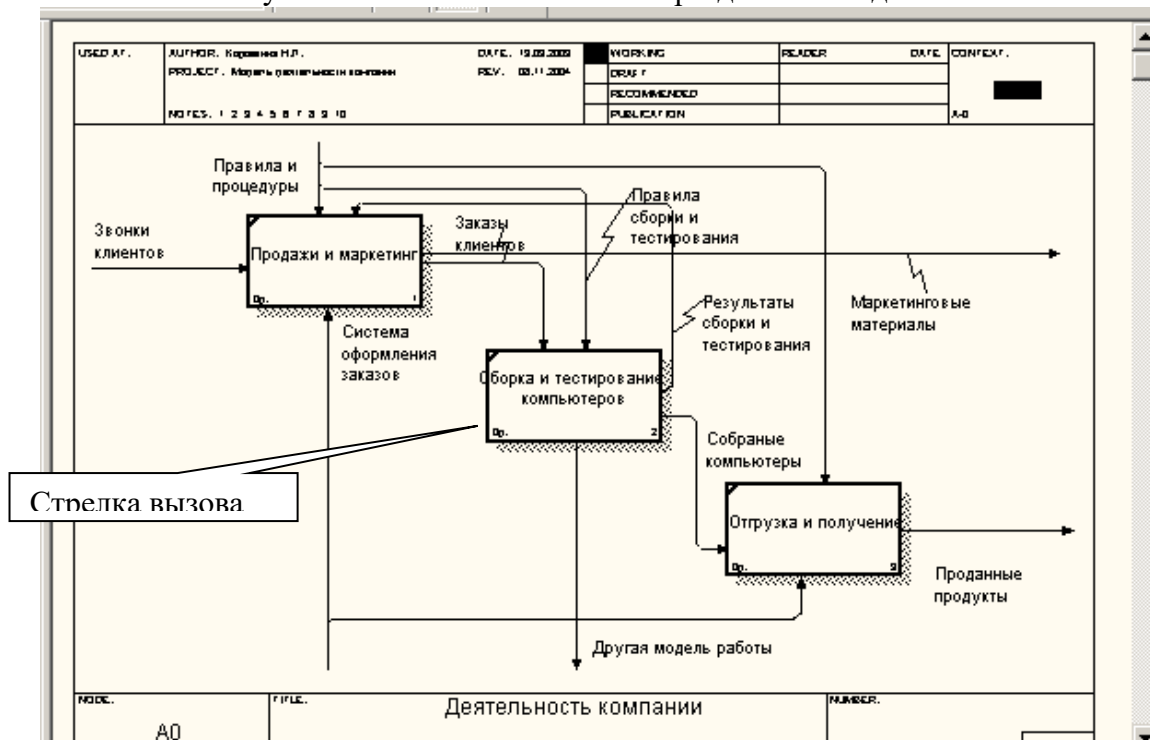



Рисунок 10. Стрелка вызова, появляющаяся при расщеплении модели

Граничные стрелки. Стрелки на контекстной диаграмме служат для описания взаимодействия системы с окружающим миром. Они могут начинаться у границы диаграммы и заканчиваться у работы, или наоборот. Такие стрелки называются граничными.

Для внесения граничной стрелки входа следует:

- щелкнуть по кнопке с символом стрелки 

в палитре инструментов перенести курсор к левой стороне экрана, пока не появится начальная штриховая полоска;

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

- вернуться в палитру инструментов и выбрать опцию редактирования стрелки



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

Стрелки управления, выхода, механизма и выхода изображаются аналогично. Имена вновь внесенных стрелок (рис.11) автоматически заносятся в словарь Arrow Dictionary

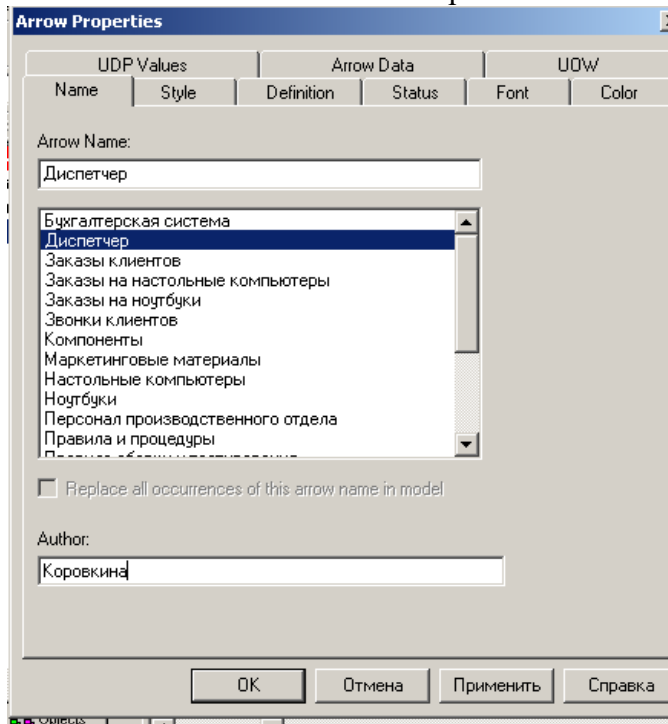


Рисунок 11. Диалог IDEF0 Arrow Properties

ICOM-коды. Диаграмма декомпозиции предназначена для детализации работы. В отличие от моделей, отображающих структуру организации, работа на диаграмме верхнего уровня в IDEF0 - это не элемент управления нижестоящими работами. Работы нижнего уровня - это то же самое, что работы верхнего уровня, но в более детальном изложении. Как следствие этого границы работы верхнего уровня - это то же самое, что границы диаграммы декомпозиции. *ICOM* (аббревиатура от *Input, Control, Output u Mechanism*) - коды, предназначенные для идентификации граничных стрелок. Код *ICOM* содержит префикс, соответствующий типу стрелки (I, C, O или M), и порядковый номер.

VPwin вносит *ICOM*-коды автоматически. Для отображения *ICOM*-кодов следует включить опцию *ICOM codes* на закладке *Display* диалога *Model Properties* (меню *Model/Model Properties*) (рис.12).

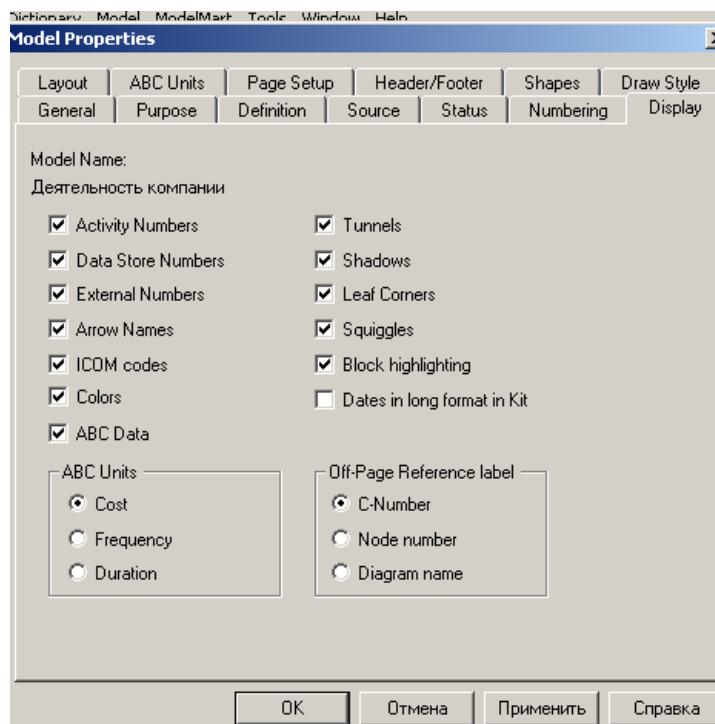


Рисунок 12. Включение опции ICOM codes на закладке Display

Словарь *стрелок* редактируется при помощи специального редактора Arrow Dictionary Editor, в котором определяется стрелка и вносится относящийся к ней комментарий (рис. 13). Словарь стрелок решает очень важную задачу. Диаграммы создаются аналитиком для того, чтобы провести сеанс экспертизы, т. е. обсудить диаграмму со специалистом предметной области. В любой предметной области формируется профессиональный жаргон, причем очень часто жаргонные выражения имеют нечеткий смысл и воспринимаются разными специалистами по-разному. В то же время аналитик - автор диаграмм должен употреблять те выражения, которые наиболее понятны экспертам. Поскольку формальные определения часто сложны для восприятия, аналитик вынужден употреблять профессиональный жаргон, а, чтобы не возникло неоднозначных трактовок, в словаре стрелок каждому понятию можно дать расширенное и, если это необходимо, формальное определение.

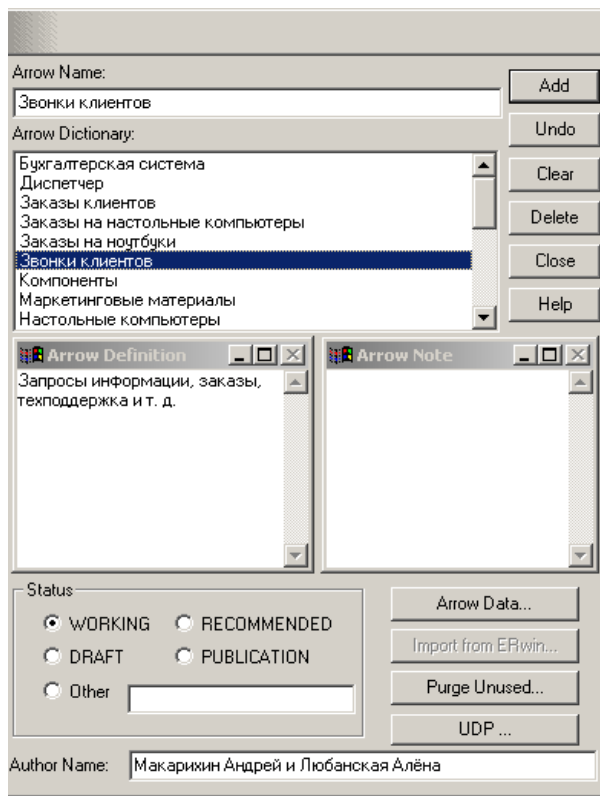


Рисунок 13. Редактирование словаря стрелок

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

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

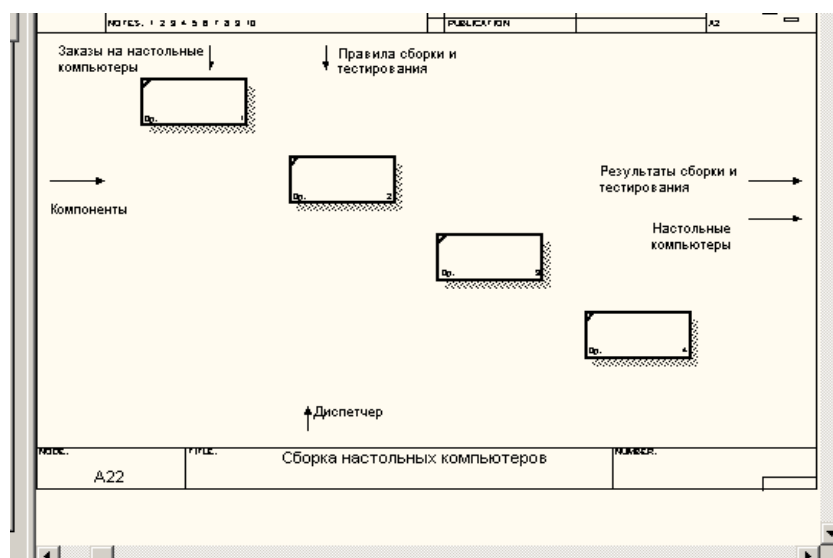


Рисунок 14. Пример несвязанных стрелок

На рис. 14 приведен фрагмент диаграммы декомпозиции с несвязанными стрелками, генерирующийся VRwin при декомпозиции работы "Сборка настольных компьютеров" (см. рис. 9). Для связывания стрелок входа, управления или механизма необходимо перейти в режим редактирования стрелок, щелкнуть по наконечнику стрелки и щелкнуть по соответствующему сегменту работы. Для связывания стрелки выхода необходимо перейти в режим редактирования стрелок, щелкнуть по сегменту выхода работы и затем по стрелке.

Внутренние стрелки. Для связи работ между собой используются внутренние стрелки, то есть стрелки, которые не касаются границы диаграммы, начинаются у одной и кончаются у другой работы.

Для рисования внутренней стрелки необходимо в режиме рисования стрелок щелкнуть по сегменту (например, выхода) одной работы и затем по сегменту (например, входа) другой. В IDEF0 различают пять типов связей работ.

Связь по входу (output-input), когда стрелка выхода вышестоящей работы (далее - просто выход) направляется на вход нижестоящей (например, на рис. 15 стрелка "Собранные компьютеры" связывает работы "Сборка и тестирование компьютеров" и "Отгрузка и получение").

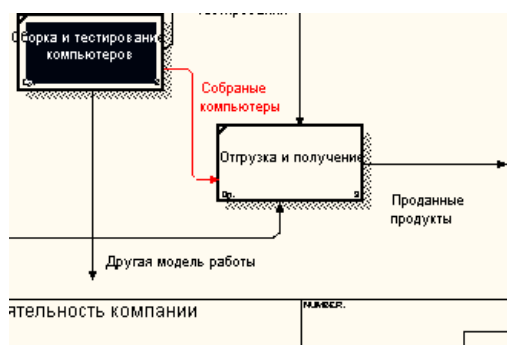


Рисунок 15. Связь по входу

Связь по управлению (output-control), когда выход вышестоящей работы направляется на управление нижестоящей. Связь по управлению показывает доминирование вышестоящей работы. Данные или объекты выхода вышестоящей работы не меняются в нижестоящей. На рис. 16 стрелка "Заказы клиентов" связывает работы "Продажи и маркетинг" и "Сборка и тестирование компьютеров".

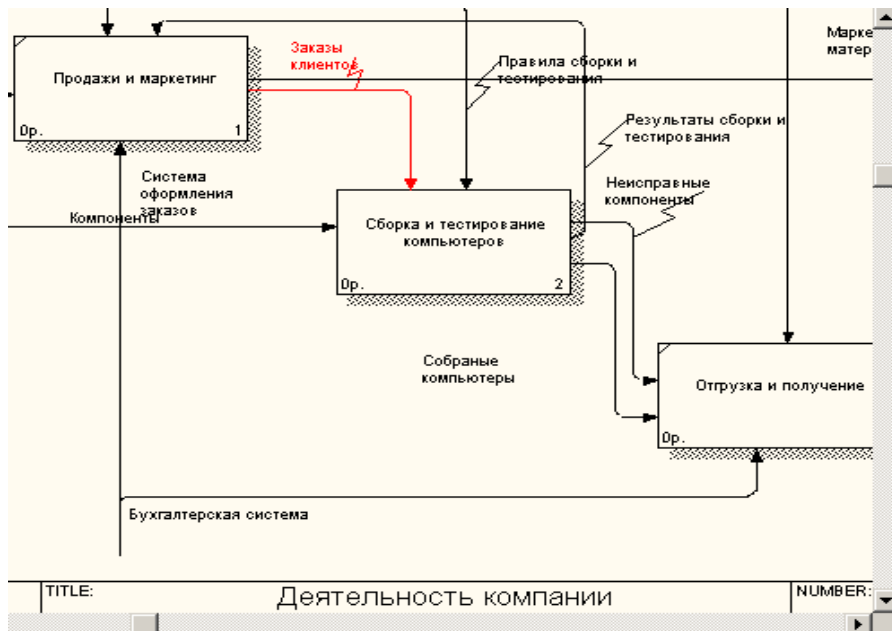


Рисунок 16. Связь по управлению

Обратная связь по входу (*output-input feedback*), когда выход нижестоящей работы направляется на вход вышестоящей. Такая связь, как правило, используется для описания циклов. На рис. 17 стрелка "Результаты тестирования" связывает работы "Тестирование компьютеров" и "Отслеживание расписания и управление сборкой и тестированием".

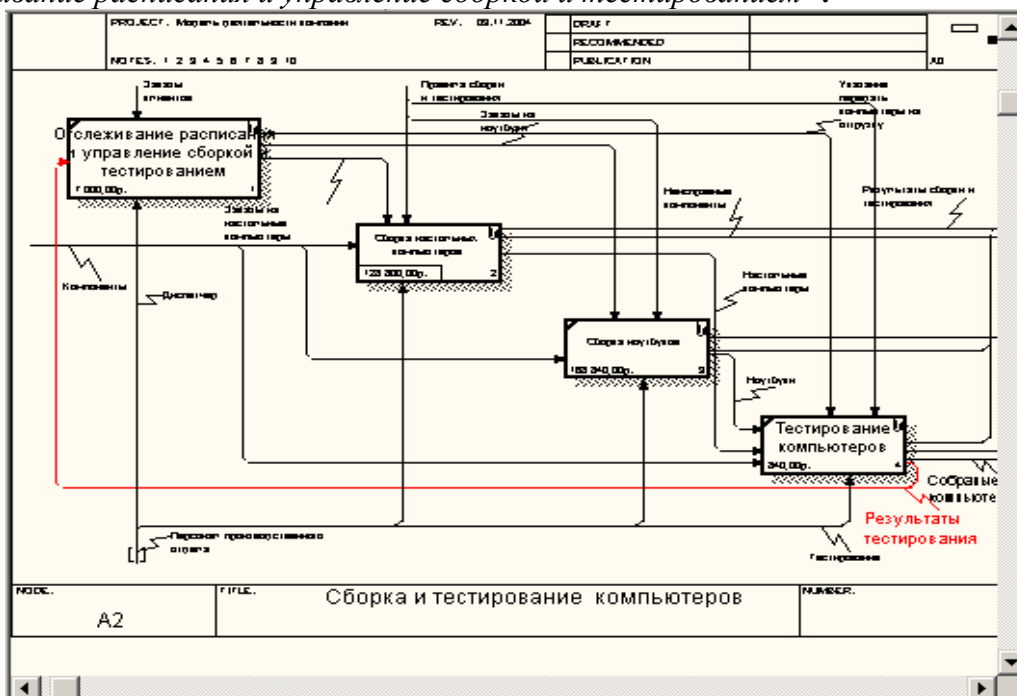


Рисунок 1 Обратная связь по входу

Обратная связь по управлению (*output-control feedback*), когда выход нижестоящей работы направляется на управление вышестоящей (стрелка "Результаты сборки и тестирования", рис. 18). Обратная связь по управлению часто свидетельствует об эффективности бизнес-процесса. На рис. 18 объем продаж может быть повышен путем непосредственного регулирования процессами сборки и тестирования компьютеров (выхода) работы "Сборка и тестирование компьютеров".

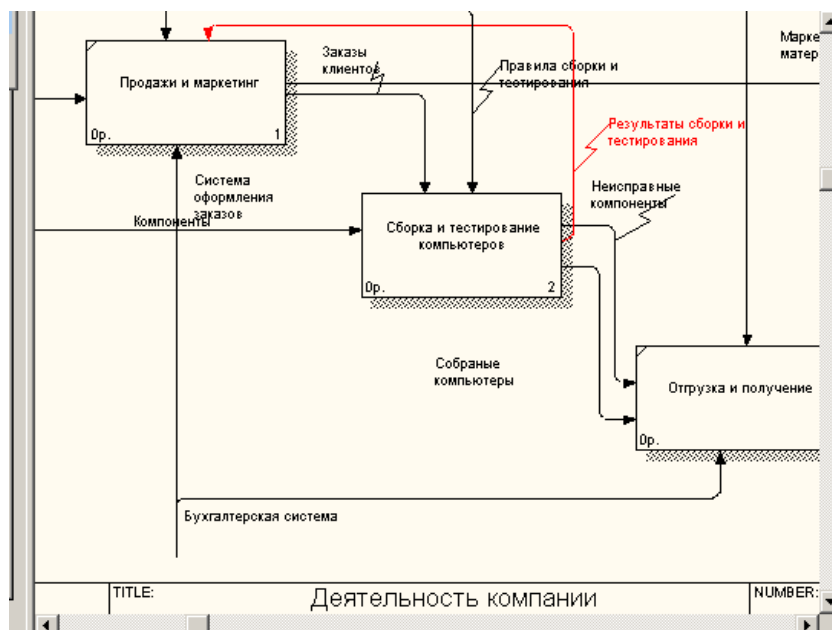


Рисунок 18. Обратная связь по управлению

Связь выход-механизм (output-mechanism), когда выход одной работы направляется на механизм другой. Эта взаимосвязь используется реже остальных и показывает, что одна работа подготавливает ресурсы, необходимые для проведения другой работы (рис. 19).

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

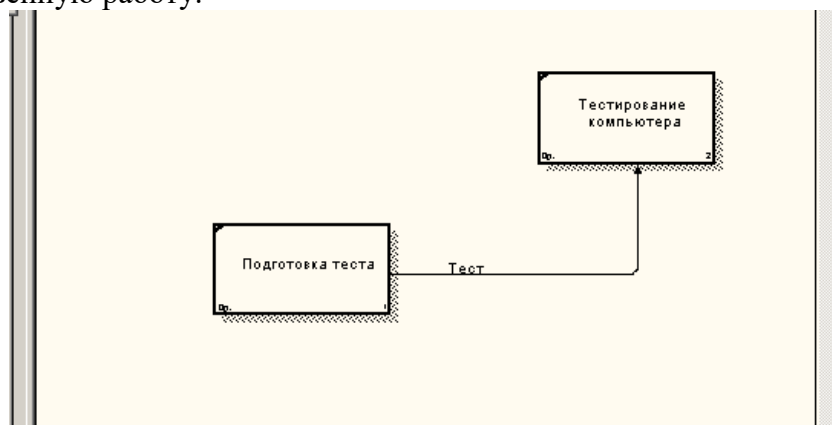


Рисунок 19. Связь выход-механизм

Разветвляющиеся и сливающиеся стрелки. Одни и те же данные или объекты, порожденные одной работой, могут использоваться сразу в нескольких других работах. С другой стороны, стрелки, порожденные в разных работах, могут представлять собой одинаковые или однородные данные или объекты, которые в дальнейшем используются или перерабатываются в одном месте. Для моделирования таких ситуаций в IDEF0 используются разветвляющиеся и сливающиеся стрелки. Для разветвления стрелки нужно в режиме редактирования стрелки щелкнуть по фрагменту стрелки и по соответствующему сегменту работы. Для слияния двух стрелок выхода нужно в режиме редактирования стрелки сначала щелкнуть по сегменту выхода работы, а затем по соответствующему фрагменту стрелки.

Смысл разветвляющихся и сливающихся стрелок передается именованием каждой ветви стрелок. Существуют определенные правила именования таких стрелок. Рассмотрим их на примере разветвляющихся стрелок. Если стрелка именована до разветвления, а после разветвления ни одна из ветвей не именована, то подразумевается, что каждая ветвь моделирует те же данные или объекты, что и ветвь до разветвления (рис. 20).

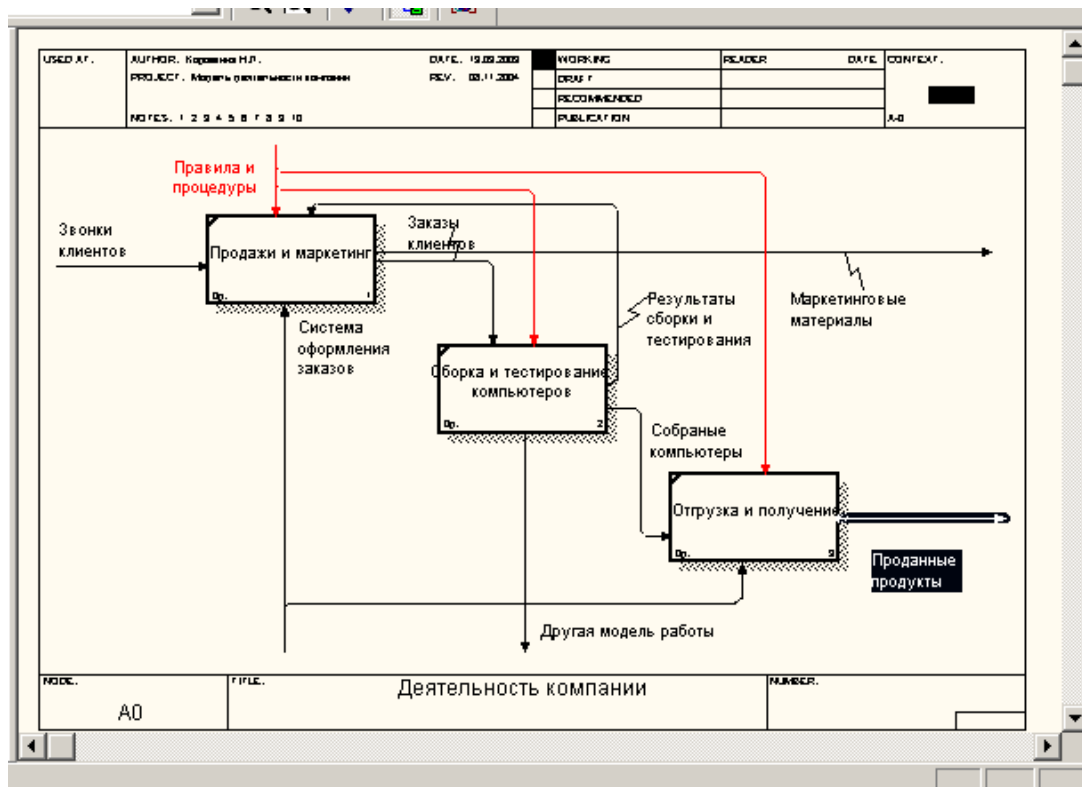


Рисунок 20. Пример именованной разветвляющейся стрелки

Если стрелка именована до разветвления, а после разветвления какая-либо из ветвей именована, то подразумевается, что эти ветви соответствуют именованию. Если при этом какая-либо ветвь после разветвления осталась неименованной, то подразумевается, что она моделирует те же данные или объекты, что и ветвь до разветвления (рис. 21).

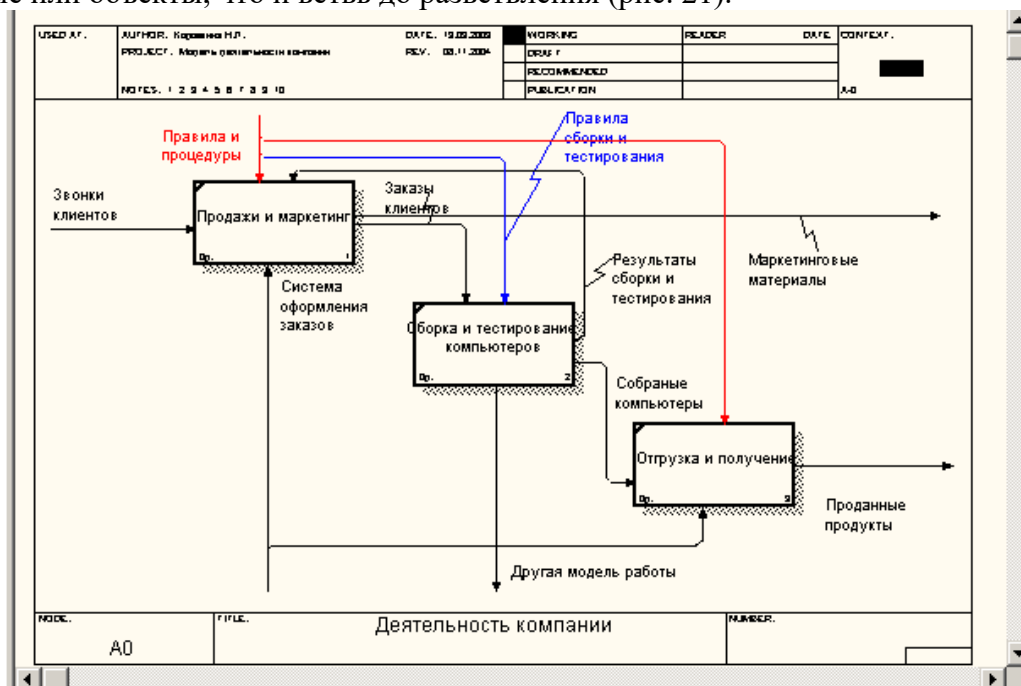


Рисунок 21. Пример именованной разветвляющейся стрелки

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

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

Туннелирование стрелок. Вновь внесенные граничные стрелки на диаграмме декомпозиции нижнего уровня изображаются в квадратных скобках и автоматически не появляются на диаграмме верхнего уровня (рис. 22).

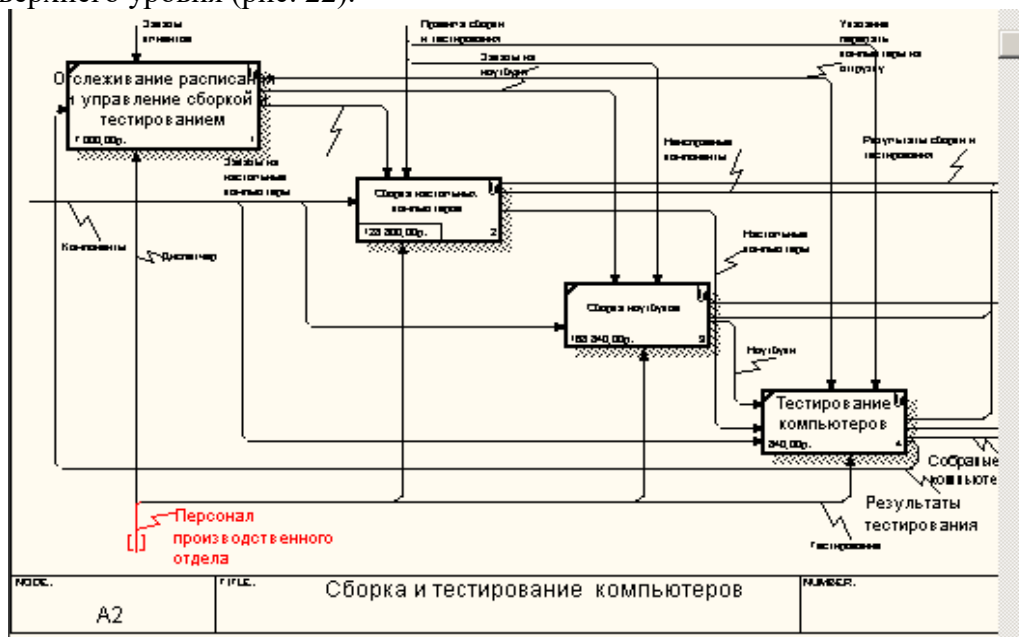


Рисунок 22. Неразрешенная (unresolved) стрелка

Для их "перетаскивания" наверх нужно щелкнуть правой кнопкой мыши по квадратным скобкам граничной стрелки и в контекстном меню выбрать команду Arrow Tunnel (рис. 23).

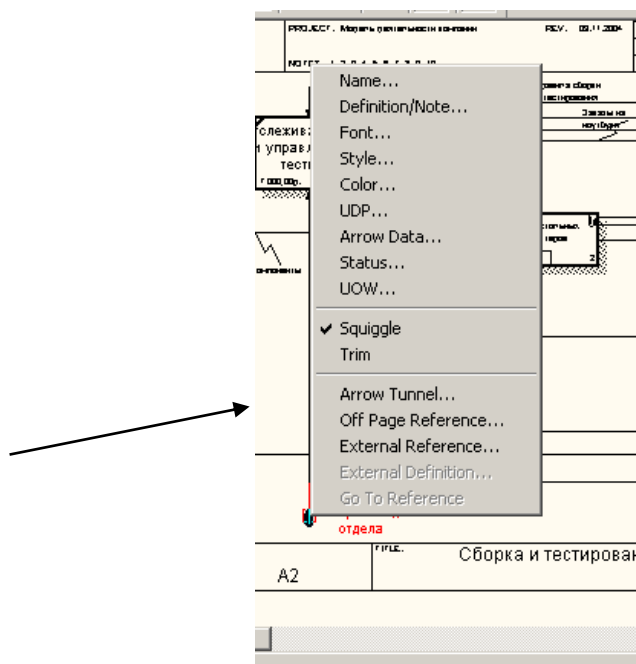


Рисунок 23. Выбор команды из контекстного меню

Появляется диалог Border Arrow Editor (рис. 24).

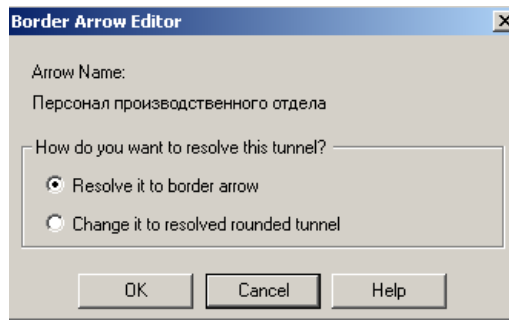


Рисунок24. Диалог Border Arrow Editor

Если щелкнуть по кнопке Resolve Border Arrow, стрелка мигрирует на диаграмму верхнего уровня, если по кнопке Change To Tunnel - стрелка будет туннелирована и не попадет на другую диаграмму. Туннельная стрелка изображается с круглыми скобками на конце (рис. 25).

Туннелирование может быть применено для изображения малозначимых стрелок. Если на какой-либо диаграмме нижнего уровня необходимо изобразить малозначимые данные или объекты, которые не обрабатываются или не используются работами на текущем уровне, то их необходимо направить на вышестоящий уровень (на родительскую диаграмму). Если эти данные не используются на родительской диаграмме, их нужно направить еще выше, и т. д. В результате малозначимая стрелка будет изображена на всех уровнях и затруднит чтение всех диаграмм, на которых она присутствует. Выходом является туннелирование стрелки на самом нижнем уровне. Такое туннелирование называется "не-в-родительской-диаграмме".

Другим примером туннелирования может быть ситуация, когда стрелка механизма мигрирует с верхнего уровня на нижний, причем на нижнем уровне этот механизм используется одинаково во всех работах без исключения. (Предполагается, что не нужно детализировать стрелку механизма, т. е. стрелка механизма на дочерней работе именована до разветвления, а после разветвления ветви не имеют собственного имени). В этом случае стрелка механизма на нижнем уровне может быть удалена, после чего на родительской диаграмме она может быть туннелирована, а в комментарии к стрелке или в словаре можно указать, что механизм будет использоваться во всех работах дочерней диаграммы декомпозиции. Такое туннелирование называется "не-в-дочерней-работе" (рис. 25).

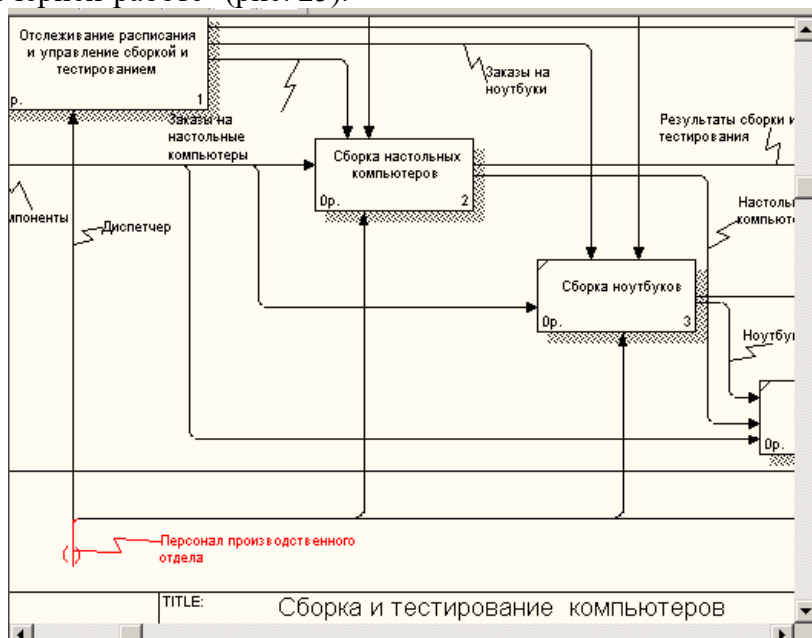


Рисунок 25. Типы туннелирования стрелок

Нумерация работ и диаграмм. Все работы модели нумеруются. Номер состоит из префикса и числа. Может быть использован префикс любой длины, но обычно используют префикс А. Контекстная (корневая) работа дерева имеет номер А0.(здесь должна быть буква О или ноль?) Работы i декомпозиции А0 (и здесь) 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 в закладке Presentation диалога Model Properties (меню Edit/Model Properties).

Диаграммы IDEF0 имеют двойную нумерацию. Во-первых, диаграммы имеют номера по узлу. Контекстная диаграмма всегда имеет номер А-0, декомпозиция контекстной диаграммы - номер А0, остальные диаграммы декомпозиции - номера по соответствующему узлу (например, А1, А2, А21, А213 и т. д.). 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 MSB00021.

Тема № 5: «Каноническое проектирование информационных систем»

Выполните моделирование в ВРWin согласно приведенному описанию.

Диаграммы дерева узлов и FEO

Диаграмма дерева узлов показывает иерархию работ в модели и позволяет рассмотреть всю модель целиком, но не показывает взаимосвязи между работами (стрелки) (рис. 26). Процесс создания модели работ является итерационным, следовательно, работы могут менять свое расположение в дереве узлов многократно. Чтобы не запутаться и проверить способ декомпозиции, следует после каждого изменения создавать диаграмму дерева узлов. Впрочем, ВРwin имеет мощный инструмент навигации по модели -Model Explorer, который позволяет представить иерархию работ и диаграмм в удобном и компактном виде, однако этот инструмент не является составляющей стандарта IDEF0.

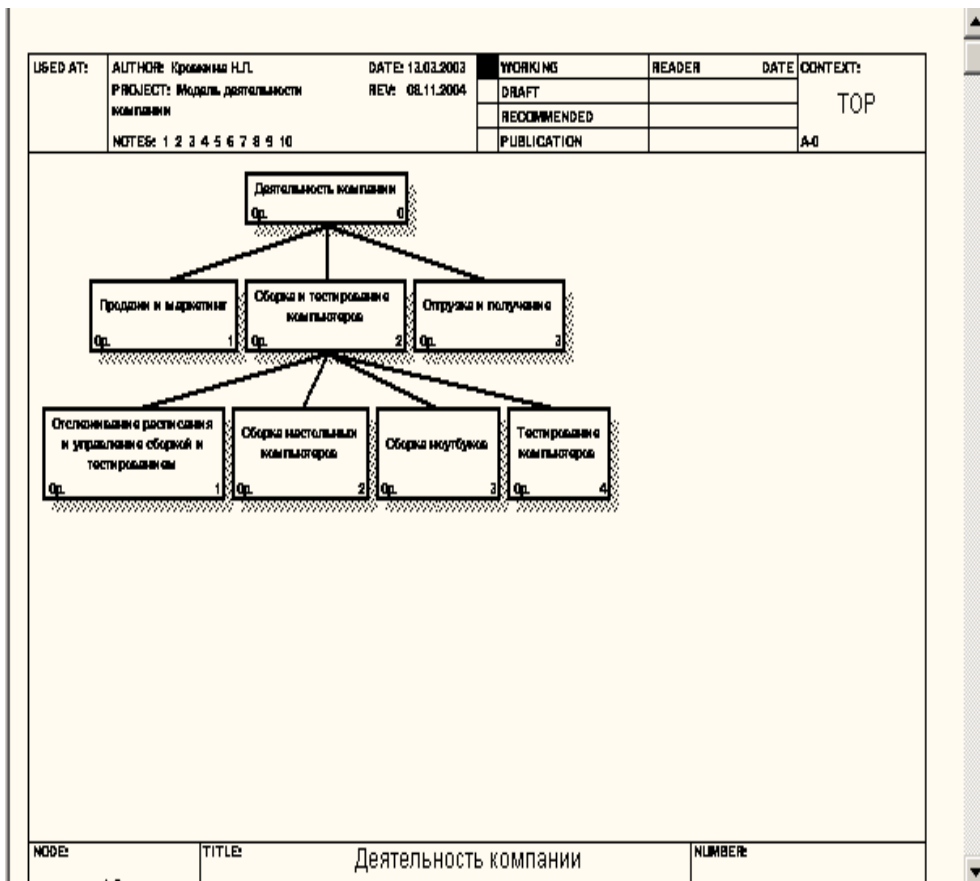


Рисунок 26. Диаграмма дерева узлов

Для создания диаграммы дерева узлов следует выбрать в меню пункт Diagram/Add Node Tree (рис. 27). Возникает диалог формирования диаграммы дерева узлов Node Tree Definition (рис. 28, 29).

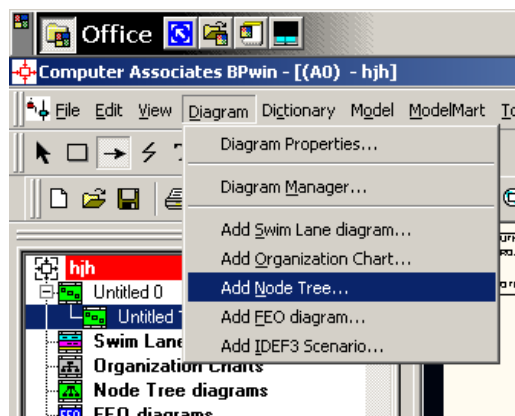


Рисунок 2 Выбор команды для формирования диаграммы дерева узлов

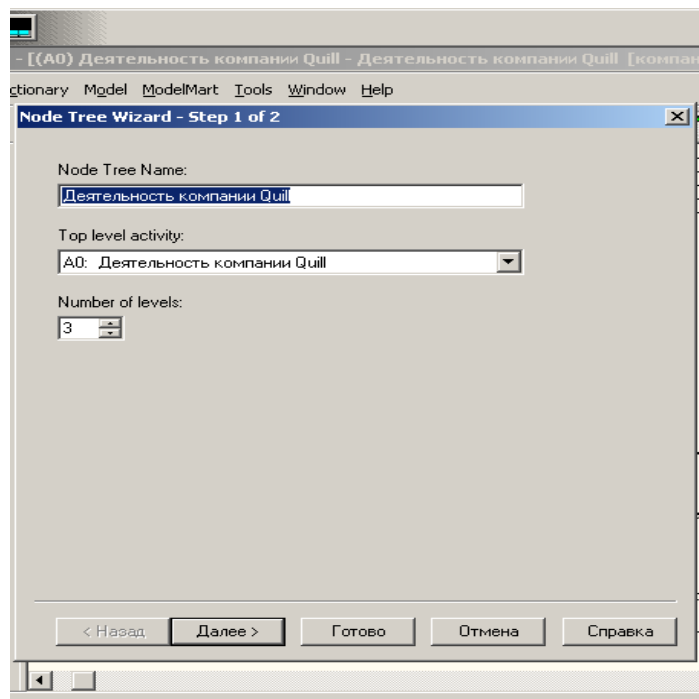


Рисунок 28. Диалог настройки диаграммы дерева узлов (шаг 1)

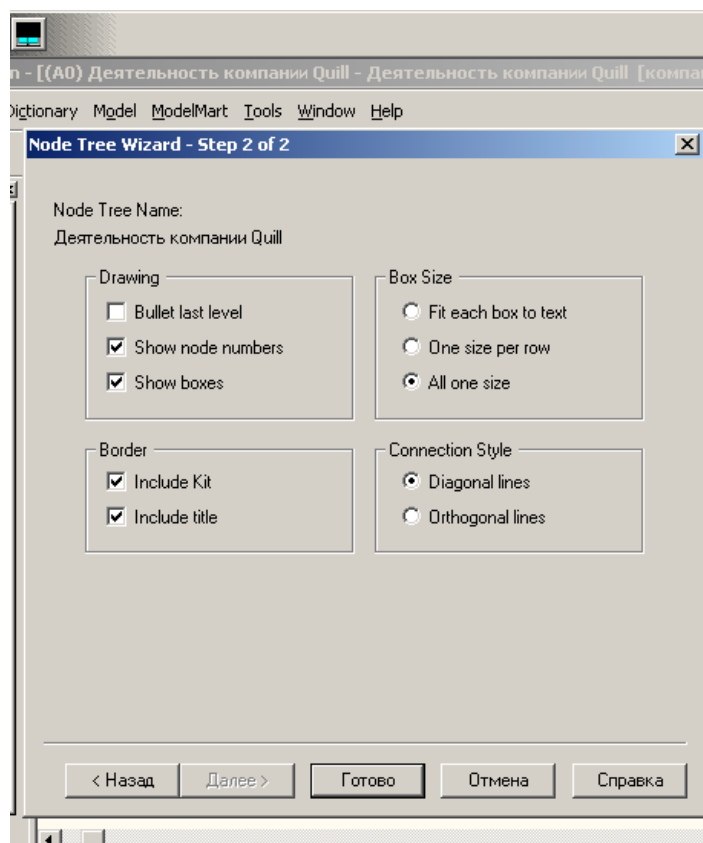


Рисунок 29. Диалог настройки диаграммы дерева узлов (шаг 2)

В диалоге Node Tree Definition следует указать глубину дерева - Number of Levels (по умолчанию 3) и корень дерева (по умолчанию - родительская работа текущей диаграммы). По умолчанию нижний уровень декомпозиции показывается в виде списка, остальные работы - в виде прямоугольников. Для отображения всего дерева в виде прямоугольников следует выключить опцию Bullet Last Level. При создании дерева узлов следует указать имя диаграммы, поскольку, если в нескольких диаграммах в качестве корня на дереве узлов использовать одну и ту же работу, все эти диаграммы получат одинаковый номер (номер узла + постфикс N, например AON) и в списке открытых диаграмм (пункт меню Window) их можно будет различить только по имени.

Диаграммы "только для экспозиции" (FEO) часто используются в модели для иллюстрации других точек зрения, для отображения отдельных деталей, которые не поддерживаются явно синтаксисом IDEF0. Диаграммы FEO позволяют нарушить любое синтаксическое правило, поскольку по сути являются просто картинками - копиями стандартных диаграмм и не включаются в анализ синтаксиса. Для создания диаграммы FEO следует выбрать пункт меню Diagram/Add FEO Diagram. В возникающем диалоге Add New FEO Diagram следует указать имя диаграммы FEO и тип родительской диаграммы (рис. 30.).

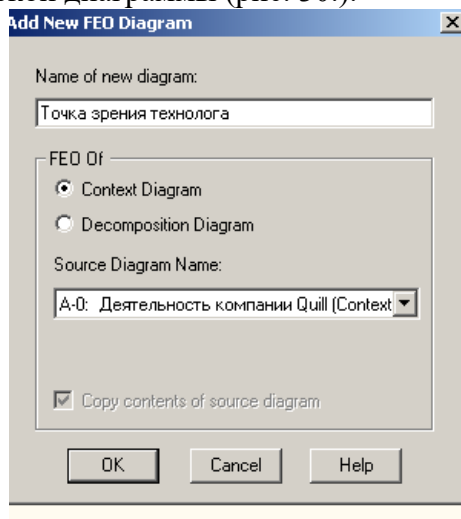


Рисунок 30. Диалог создания FEO-диаграммы

Новая диаграмма получает номер, который генерируется автоматически (номер родительской диаграммы по узлу + постфикс F, например A1F).

Каркас диаграммы

На рис. 31 показан типичный пример диаграммы декомпозиции с граничными рамками, которые называются каркасом диаграммы.

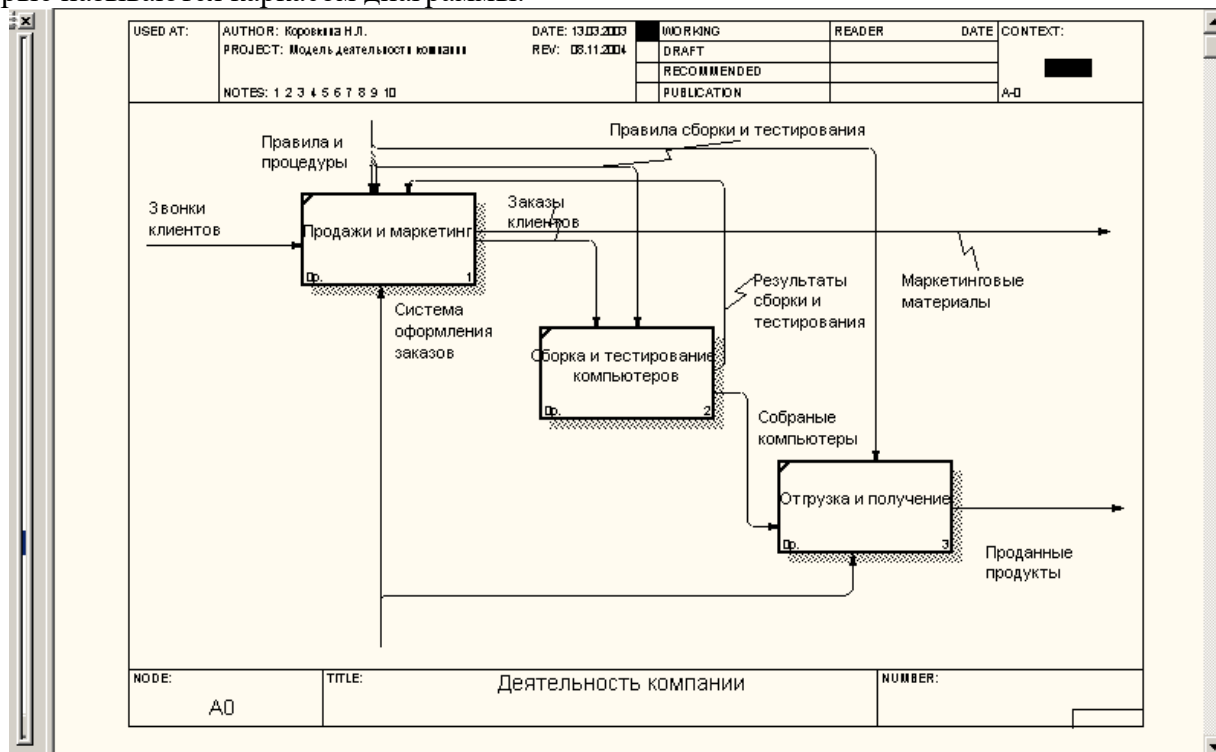


Рисунок 31. Пример диаграммы декомпозиции с каркасом

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

Смысл элементов каркаса приведен в табл. 1 и 2.

Таблица 1. Поля заголовка каркаса (слева направо)

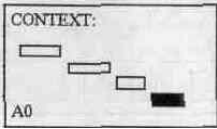
Поле	Смысл
Used At	Используется для указания на родительскую работу в случае, если на текущую диаграмму ссылались посредством стрелки вызова
Autor, Date, Rev, Prpject	Имя создателя диаграммы, дата создания и имя проекта, в рамках которого была создана диаграмма. REV-дата последнего редактирования диаграммы
Notes 123456789 10	Используется при проведении сеанса экспертизы. Эксперт должен (на бумажной копии диаграммы) указать число замечаний, вычеркивая цифру из списка каждый раз при внесении нового замечания
Status	Статус отображает стадию создания диаграммы, отображая все этапы публикации
Working	Новая диаграмма, кардинально обновленная диаграмма или новый автор диаграммы
Draft	Диаграмма прошла первичную экспертизу и готова к дальнейшему обсуждению
Recommended	Диаграмма и все ее сопровождающие документы прошли экспертизу. Новых изменений не ожидается
Publication	Диаграмма готова к окончательной печати и публикации
Reader	Имя читателя (эксперта)
Date	Дата прочтения (экспертизы)
Context	<p>Схема расположения работ в диаграмме верхнего уровня. Работа, являющаяся родительской, показана темным прямоугольником, остальные – светлым. На контекстной диаграмме (A-0) показана надпись TOP. В левом нижнем углу показывается номер по узлу родительской диаграммы:</p> 

Таблица 2. Поля подвала каркаса (слева направо)

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

Значения полей каркаса задаются в диалоге Diagram Properties (меню Diagram /Diagram Properties) - рис. 32.

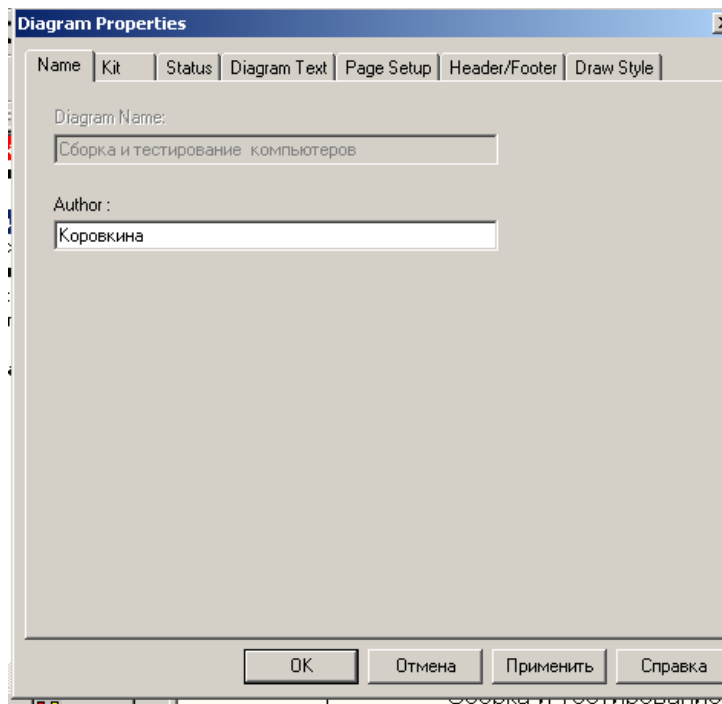


Рисунок 32. Диалог Diagram Properties

Слияние и расщепление моделей

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

ВРwin использует для слияния и разветвления моделей стрелки вызова. Для слияния необходимо выполнить следующие условия:

- Обе сливаемые модели должны быть открыты в Врwin.
- Имя модели-источника, которое присоединяют к модели-цели, должно совпадать с именем стрелки вызова работы в модели-цели.
- Стрелка вызова должна исходить из недекомпозируемой работы (работа должна иметь диагональную черту в левом верхнем углу) (рис. 33).
- Имена контекстной работы подсоединяемой модели-источника и работы на модели-цели, к которой мы подсоединяем модель-источник, должны совпадать.
- Модель-источник должна иметь, по крайней мере, одну диаграмму декомпозиции.

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

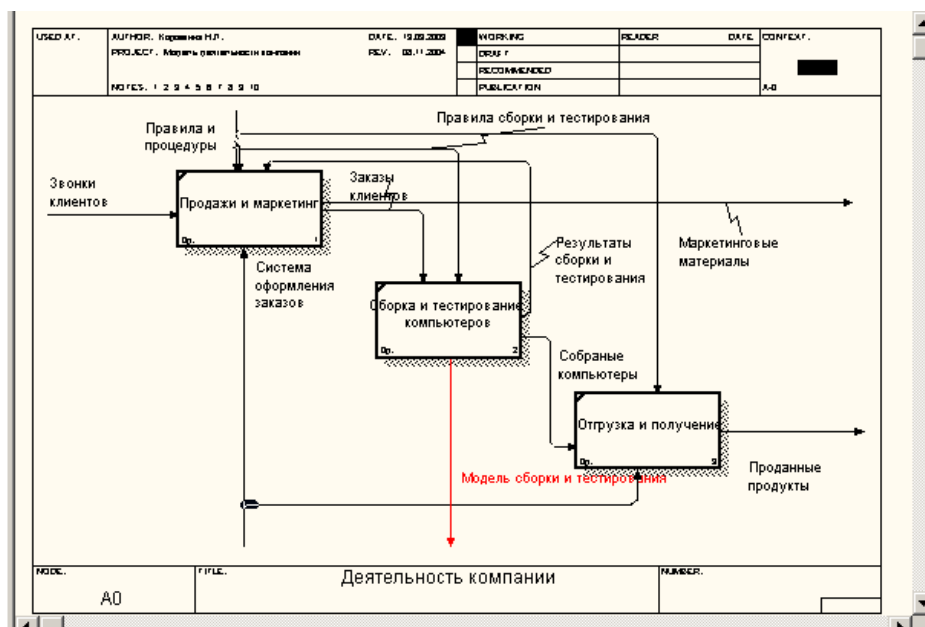


Рисунок 33. Стрелка вызова работы "Сборка и тестирование компьютеров" модели-цели

Появляется диалог, в котором следует указать опции слияния модели (рис. 34). При слиянии моделей объединяются и словари стрелок и работ. В случае одинаковых определений возможна перезапись определений или принятие определений из модели-источника. То же относится к именам стрелок, хранилищам данных и внешним ссылкам. (Хранилища данных и внешние ссылки - объекты диаграмм потоков данных, DFD, будут рассмотрены ниже.)

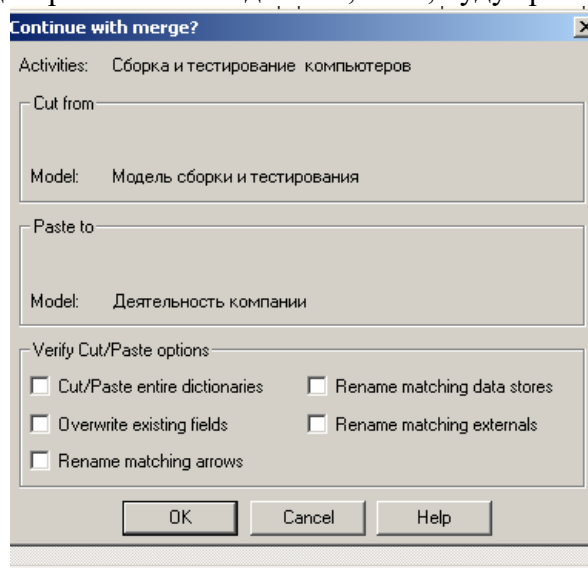


Рисунок 34. Диалог Continue with merge

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

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

Разделение моделей производится аналогично. Для отщепления ветви от модели следует щелкнуть правой кнопкой мыши по декомпозированной работе (работа не должна иметь диагональной черты в левом верхнем углу) и выбрать во всплывающем меню пункт Split Model. В появившемся диалогe Split Options следует указать имя создаваемой модели. После

подтверждения расщепления в старой модели работа станет недекомпозированной (признак - диагональная черта в левом верхнем углу), будет создана стрелка вызова, ее имя будет совпадать с именем новой модели, и, наконец, будет создана новая модель, причем имя контекстной работы будет совпадать с именем работы, от которой была "оторвана" декомпозиция.

Тема № 6: «Автоматизированное проектирование ИС»

Цель работы: ознакомиться с основными понятиями процесса моделирования данных, системой обозначений методологии IDEF1x, интерфейсом ERWIN и применить полученные знания для разработки простейших моделей данных.

Результат обучения. После обучения студент должен:

- знать основные понятия процесса моделирования данных;
- знать систему обозначений методологии IDEF1x;
- уметь применять знания для разработки простейших моделей данных;
- знать основные понятия методологии функционального моделирования IDEF1x.

Установите триал версию программы AllFusion ERwin Data Modeler или ERWin.

Выполните моделирование в ERWin согласно приведенному описанию.

1. Общие понятия о моделировании данных

Моделирование данных – это процесс описания информационных структур и бизнес-правил для определения потребностей инфокоммуникационной системы.

При проектировании базы данных необходимо провести анализ целей этой системы и выявить требования к ней. Сбор данных начинается с изучения сущностей предметной области, процессов, использующих эти сущности, и связей между ними и заканчивается построением концептуальной модели, являющейся базисом формализованной модели данных.

Концептуальная модель служит, средством общения между различными пользователями и поэтому разрабатывается без учета особенностей физического представления данных. При проектировании концептуальной модели все усилия разработчика должны быть направлены в основном на структуризацию данных и выявление взаимосвязей между ними без рассмотрения особенностей реализации и вопросов эффективности обработки. Концептуальное проектирование является центральной частью, ядром всего процесса проектирования баз данных. Самой популярной моделью концептуального проектирования является модель «сущность-связь» (ER-модель), она относится к семантическим моделям.

Концептуальная модель носит декларативный характер, по сути, она является базой знаний об объекте исследований и служит для формализации представления объекта (задачи) и анализа требований к системе его моделирования (решения). Декларативный характер концептуальной модели не накладывает ограничений на последующий выбор типа модели для ее информационной реализации. Цель концептуальной модели - создать точное и полное отображение исследуемой области как системы, используемой в качестве источника информации для процедуры оценки ее состояния.

Перед тем, как строить ER-модель с помощью CASE- средства, следует тщательно подойти к анализу предметной области и выделению основных ее объектов.

При проектировании больших и сложных систем одной из проблем является ограниченность человеческих возможностей относительно охвата всех взаимодействий, возникающих при этом состоянии системы. Следовательно, наиболее эффективным способом построения таких систем является моделирование сложной системы из небольшого количества крупных частей, каждая из которых, в свою очередь, строится из частей меньшего размера до тех пор, пока самые небольшие части можно будет составить из имеющегося материала. Данный подход известен как иерархическая декомпозиция. По отношению к проектированию сложной системы это означает, что ее необходимо декомпонировать (от лат. “разделение”, “выделение”) на небольшие подсистемы, каждую из которых можно разрабатывать независимо от других. Это позволяет при мониторинге систем любого уровня сложности отслеживать информацию о состоянии системы в целом, а также по каждой из ее подсистем в отдельности.

Разработку концептуальной модели следует начинать с выявления основных сущностей и их описательных характеристик (атрибутов). Каждая сущность должна соответствовать некоторому объекту (или группе объектов) предметной области, о котором в системе будет накапливаться информация. Графически, концептуальная модель на рассматриваемом уровне разработки может быть представлена как иерархическая схема (рис.1).

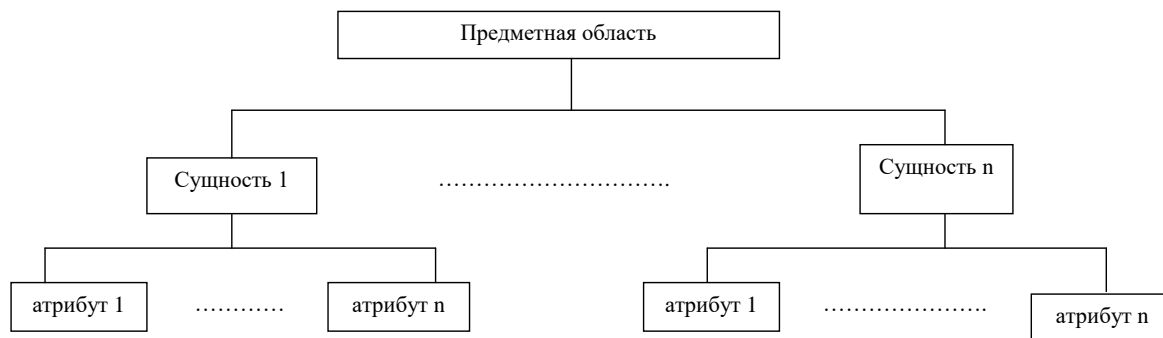


Рис. 1. Концептуальная модель предметной области

Уровни моделирования, представленные ниже, хорошо приспособлены для использования на протяжении всего жизненного цикла разработки базы данных (рис. 2). Другими словами, нужные уровни детализации создаются на каждом конкретном этапе развития проекта.

Самые общие уровни модели можно разделить на два вида. *Диаграмма зависимостей сущностей* (Entity Relationship Model, ERD) определяет основные бизнес-объекты и их взаимосвязи. *Модель, основанная на ключах* (КВ), определяет рамки требований бизнес информации и начинает раскрывать подробности.

Низкие уровни модели можно также разделить на два вида. *Полностью определенная* (Fully Attributed, FA) модель – является третьим видом модели, который содержит все детали для конкретной реализации. *Модель трансформации* (Transformation Model, TM) представляет собой трансформацию модели зависимостей в структуру соответствующую DBMS.

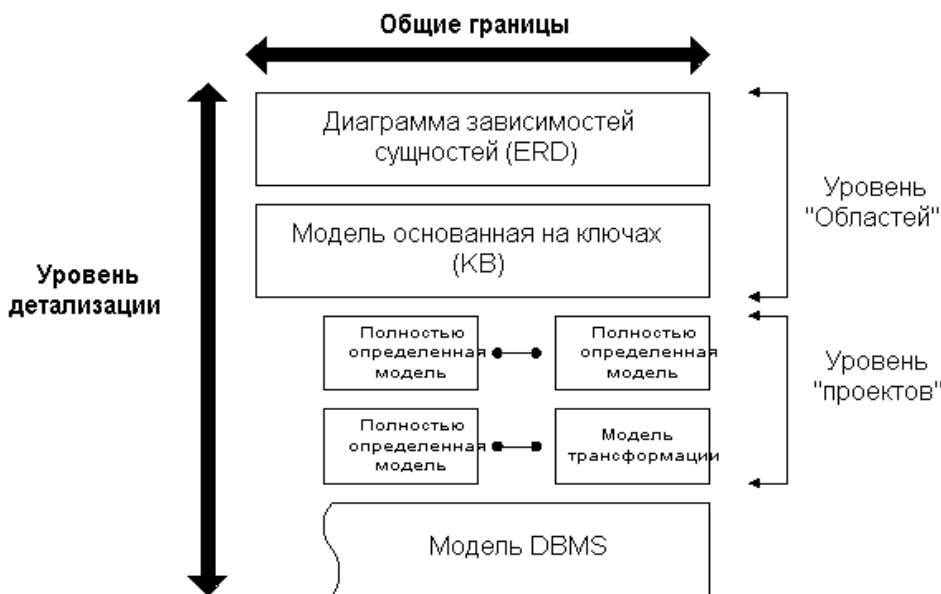


Рис. 2. Уровни проектирования базы данных IDEF1X

Существует три уровня логических моделей, используемые для охвата требований бизнес информации: *Диаграмма зависимостей сущностей* (ERD), *Модель основанная на ключах* (Key Based, KB) и *Полностью определенная модель* (FA). ERD и KB модели также называются «моделями данных области» потому, что охватывают обширные области бизнеса. Полной

противоположностью является модель FA – «модель данных проекта» потому, что она, как правило, описывает только часть всей структуры информации.

Диаграмма зависимостей сущностей является моделью данных высокого (более общего) уровня. Основной задачей Диаграммы зависимостей сущностей является обзор требований к бизнес-информации, достаточной для планирования разработки информационной системы. Эти модели не являются очень детализированными (в них включены только основные сущности), и почти отсутствуют атрибуты. Разрешены отношения многие-ко-многим, а ключи, в основном, не включаются. Одним словом, ERD является презентационной моделью, удобной для обсуждения.

KB-модель описывает основные структуры данных, которые охватывают обширные области бизнеса. Все сущности и первичные ключи включены вместе с примерами атрибутов. Основной целью модели, основанной на ключах, является широкий обзор структур данных и ключей, нужных для поддержки определенной области. Эта модель определяет контекст, в котором могут созданы подробные модели, пригодные для конкретного воплощения. Модель показывает ту же область что и область ERD, но, вместе с тем, отображает больше деталей. Полностью определенная модель – это модель данных в третьей нормальной форме, которая включает в себя сущности, атрибуты и зависимости, требуемые для конкретного проекта.

Существует два уровня физических моделей: *Модель трансформации* и *DBMS модель*. Физические модели отображают всю информацию, нужную разработчикам системы для воплощения логической модели в систему БД. Модель трансформации является также «моделью данных проекта», описывающей отдельную часть всей структуры данных, предназначенную для обеспечения конкретного участка автоматизации. ERwin поддерживает индивидуальные проекты в области бизнеса, позволяющие разработчику разделять область с более общей моделью на подобласти, называемые тематическими областями (subject areas).

Основными задачами модели трансформации являются: обеспечение администратора базы данных (DBA) информацией, нужной для создания рациональной физической базы данных, а также предоставление контекста для определений и записей в словаре данных и записей, образующих базу данных.

Эта модель может также предоставить возможность для сравнения проекта физической базы данных и изначальных требований к бизнес информации, а также для оценки и корректировки расширяемости и ограничений базы данных.

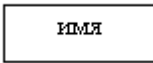
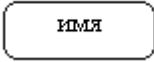
Модель трансформации напрямую переводится в DBMS модель, которая, в свою очередь, получает определения объектов физической базы данных в схеме RDBMS или каталоге базы данных. ERwin напрямую поддерживает эту модель с функцией генерации схемы. Первичные ключи становятся уникальными индексами. Альтернативные ключи и инверсные вхождения (Inversion Entries, IE) также могут стать индексами.

2. Язык моделирования данных IDEF1X

Процесс моделирования в ERwin базируется на методологии проектирования реляционных баз данных IDEF1X. Она определяет стандарты терминологии и графического изображения типовых элементов на ER-диаграммах. IDEF1X является методом для разработки реляционных баз данных и использует условный синтаксис, специально разработанный для удобного построения концептуальной схемы. Эта методология предлагает следующий графический язык для построения ER-диаграмм (табл. 1, 2,3).

Таблица 1

Обозначения сущностей в нотации IDEF1X

Элемент диаграммы	Обозначает
	независимая сущность
	зависимая сущность

Сущность в IDEF1X описывает собой совокупность или набор экземпляров похожих по свойствам, но однозначно отличаемых друг от друга по одному или нескольким признакам.

Каждый экземпляр является реализацией сущности. Таким образом, сущность в IDEF1X описывает конкретный набор экземпляров реального мира, в отличие от сущности в IDEF1, которая представляет собой абстрактный набор информационных отображений реального мира.

Список атрибутов приводится внутри прямоугольника, обозначающего сущность. Атрибуты, составляющие ключ сущности, группируются в верхней части прямоугольника и отделяются горизонтальной чертой.

Таблица 2


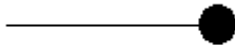
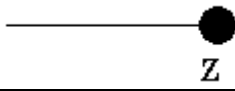
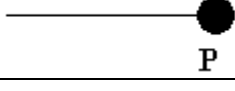
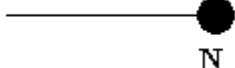
Обозначения связей в нотации IDEF1X

Элемент диаграммы	Обозначает
	идентифицирующая связь
	неидентифицирующая связь

Связи в IDEF1X представляют собой ссылки, соединения и ассоциации между сущностями. Связи это суть глаголы, которые показывают, как соотносятся сущности между собой.

Таблица 3

Обозначения кардинальности связей в нотации IDEF1X

Элемент диаграммы	Обозначает
	1,1
	0,M
	0,1
	1,M
	точно N (N - произвольное число)

3. Введение в AllFusion ERwin Data Modeler

AllFusion ERwin Data Modeler (ERwin) - CASE-средство для проектирования и документирования баз данных, которое позволяет создавать, документировать и сопровождать базы данных, хранилища и витрины данных. ERwin позволяет наглядно отображать сложные структуры данных. Удобная в использовании графическая среда упрощает разработку базы данных и автоматизирует множество трудоемких задач, уменьшая сроки создания высококачественных и высокопроизводительных транзакционных баз данных и хранилищ данных. ERwin поддерживает стандартную нотацию IDEF1x для ER-диаграмм моделей данных, нотацию IE и специальную нотацию, предназначенную для проектирования хранилищ данных - *Dimensional*.

Основное окно программы содержит следующие части: область модели данных или рабочая область; панели инструментов; браузер элементов модели (рис. 3).

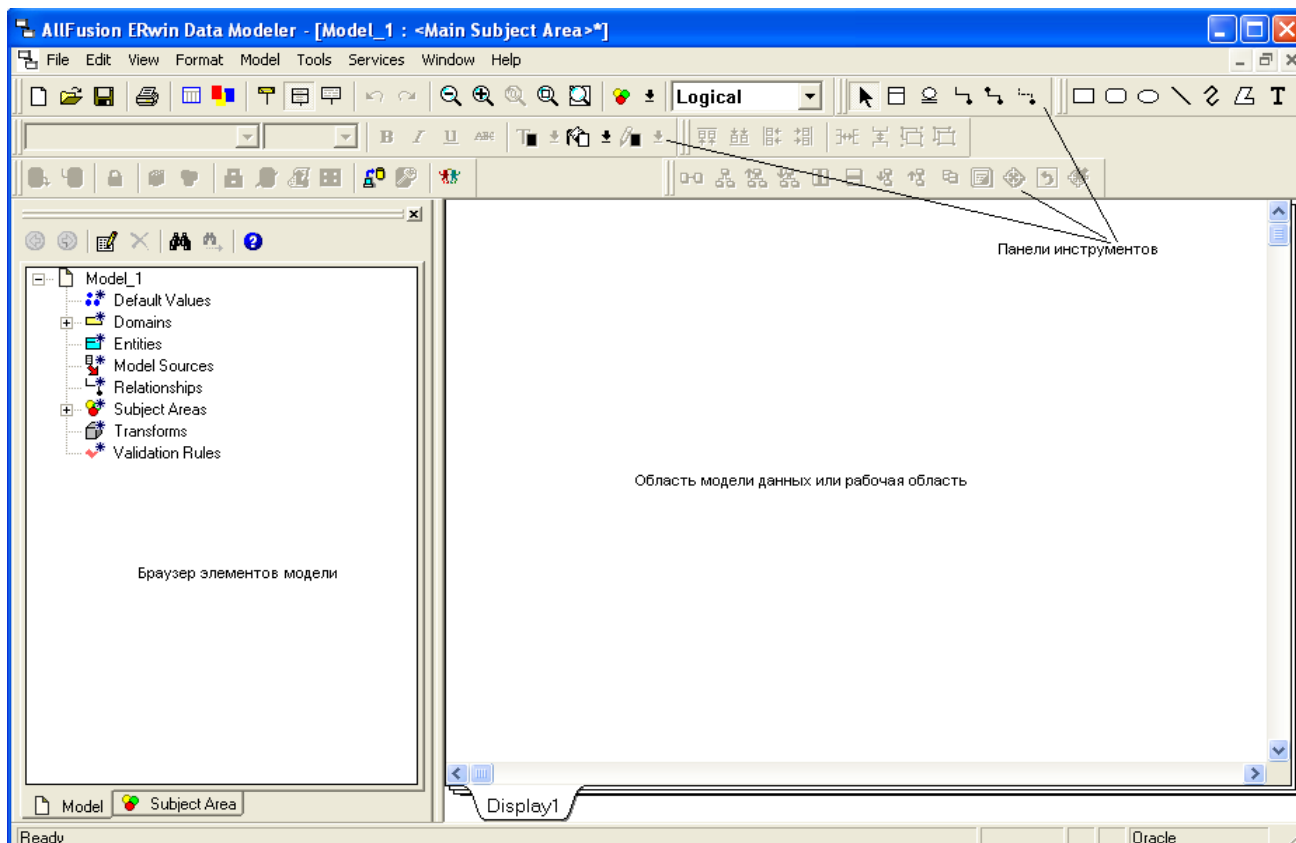


Рис. 3. Основное окно программы

Область модели данных – содержит собственно разрабатываемую модель. Если создана новая модель – это окно пустое, если открыта существующая – содержит сущности и связи между ними.

Панели инструментов: эти панели содержат практически все используемые при работе элементы. По умолчанию все панели отображаются на экране. При необходимости пользователь может отключить или, наоборот, включить требуемые модели, используя меню *View – Toolbars*.

Браузер элементов модели помогает ориентироваться среди множества элементов модели, выделяя и группируя их. Имеет вкладки *-Model, Subject Areas*.

Вкладка *Model* объединяет все элементы. Здесь можно просмотреть список всех сущностей, связей, узнать, какая сущность с какой связана, и каким отношением (родительская/дочерняя), просмотреть созданные правила проверки значений, типы данных и т.д.

Вкладка *Subject Areas* содержит список предметных областей. Предметная область – это выборка из полной модели отдельных сущностей и их областей только тех, которые относятся к заданной области.

Процесс проектирования в ERwin состоит из двух этапов – логического моделирования и физического проектирования. Переключение между ними осуществляется с помощью переключателя, расположенного на стандартной панели (рис. 4,5).

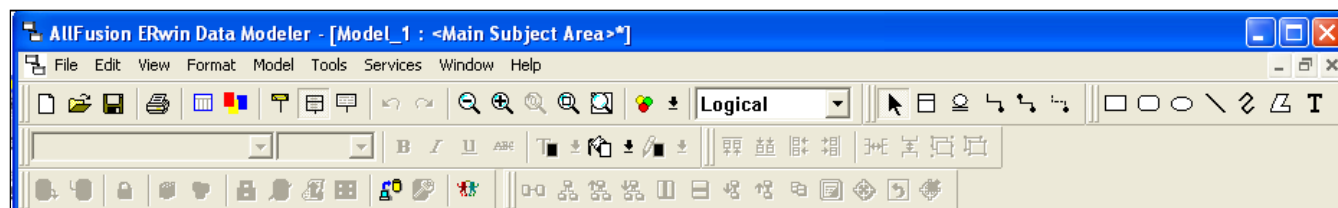


Рис. 4. Меню и инструменты AllFusion ERwin Data Modeler на логическом уровне

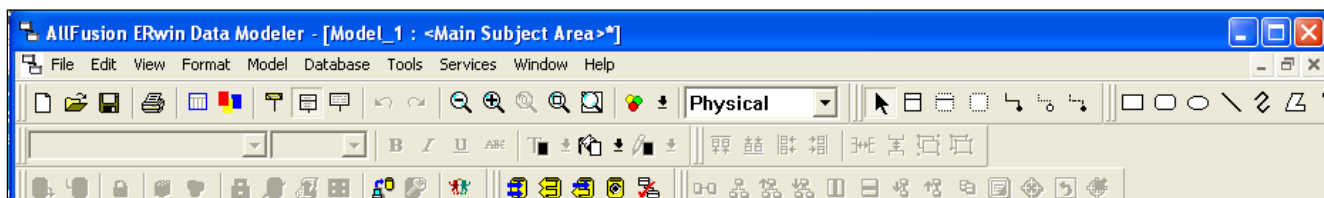
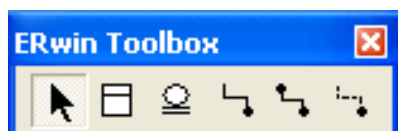
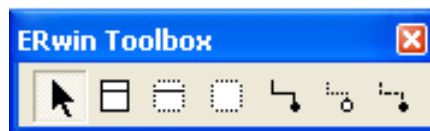


Рис. 5. Меню и инструменты AllFusion ERwin Data Modeler на физическом уровне

Основной рабочей панелью является *Toolbox*. *Toolbox* - это специальное окно, содержащее инструменты для редактирования модели. Его структура зависит от выбранной методологии проектирования и текущего уровня модели (рис. 6).



(a)



(б)

Рис. 6. Панели *Toolbox* для логического (a) и физического (б) уровней модели

Характеристика элементов панели представлена в таблице 4.

Таблица 4

Описание инструментов панели *Toolbox*

Логический уровень		
	<i>Select</i>	выбор объектов в модели
	<i>Entity</i>	создание новой сущности (на физическом уровне - таблиц)
	<i>Identifying relationship</i>	создание идентифицирующей связи «один-ко-многим»
	<i>Non-identifying relationship</i>	создание неидентифицирующей связи «один-ко-многим»
	<i>Many-to-many relationship</i>	создание связи «многие-ко-многим» (только на логическом уровне)
	<i>Complete sub-category</i>	добавление в модель новой категории сущностей (связи подтипа)
Физический уровень		
	<i>View table</i>	создание сущности-представления
	<i>View relationship</i>	создание связи для сущности-представления
	<i>Non-identifying relationship</i>	создание неидентифицирующей связи «один-ко-многим»

Сущность-представление – это виртуальная сущность, получаемая выборкой атрибутов из существующих сущностей. В терминах баз данных соответствует запросу. Использование этих инструментов производится также как в логической модели.

4. Начало работы с ERwin

ERwin имеет очень удобный пользовательский интерфейс, позволяющий представить базу данных в самых различных аспектах. После запуска ERwin необходимо создать новую диаграмму посредством меню *File* → *New*.

Далее предлагается выбрать тип создаваемой модели: только логическая / только физическая / включающая логическую и физическую (рис. 7). После этого пользователь получает возможность формировать структуру будущей базы данных с использованием средств пакета.

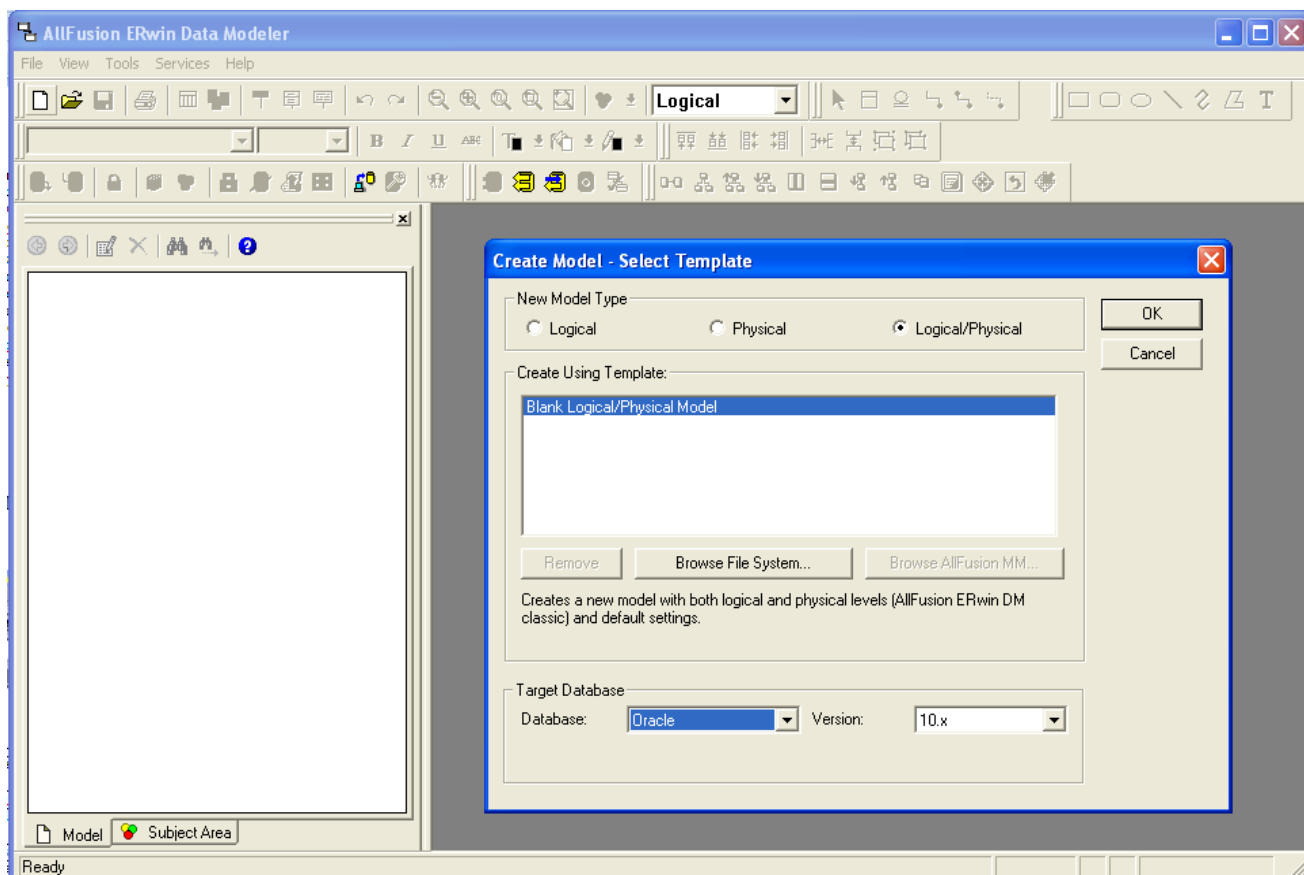


Рис. 7. Окно выбора типа модели базы данных

В любой момент можно открыть уже имеющуюся диаграмму командой *File* → *Open*, после которой нужно в диалоговом окне *ERwin Open* выбрать тот файл, который необходимо открыть. Кроме того, *ERwin* выдает список из четырех файлов, которые открывались последними (он находится в нижней части меню *File*).

Выбор типа разрабатываемой модели обусловлен тем, что обычно создание модели базы данных состоит из двух этапов: составление логической модели и создание на ее основе физической модели. *ERwin* полностью поддерживает такой процесс, он имеет два представления модели: логическое (*logical*) и физическое (*physical*). Таким образом, разработчик может строить логическую модель базы данных, не задумываясь над деталями физической реализации, т.е. уделяя основное внимание требованиям к информации и бизнес-процессам, которые будет поддерживать будущая база данных.

В процессе разработки моделей *ERwin* предлагает использовать большое количество специализированных редакторов, настраивающих ее отдельные аспекты. Однако при разработке простых моделей обходится и без них. Для разработки таких моделей достаточно лишь окна инструментов и инспектора атрибутов.

Для создания сущности надо выбрать соответствующую кнопку, после чего щелкнуть мышью на свободном месте. Создастся новая сущность, при этом активизируется ввод ее имени. После ввода имени и нажатия *Enter* выделение переходит в часть ввода ключевых атрибутов. Имена атрибутов также набираются текстом. После ввода имени ключевого атрибута можно либо клавишей *Enter* добавить еще один ключевой атрибут, либо клавишей *Tab* перейти к вводу неключевых атрибутов (рис. 8).

Для создания связи между сущностями надо выбрать требуемый тип связи и щелкнуть последовательно на родительской и дочерней сущности. Для связей «многие-ко-многим» порядок

щелканья безразличен. При создании связи «один-ко-многим» ключевые атрибуты родительской таблицы переходят в качестве внешних ключей в дочернюю таблицу.

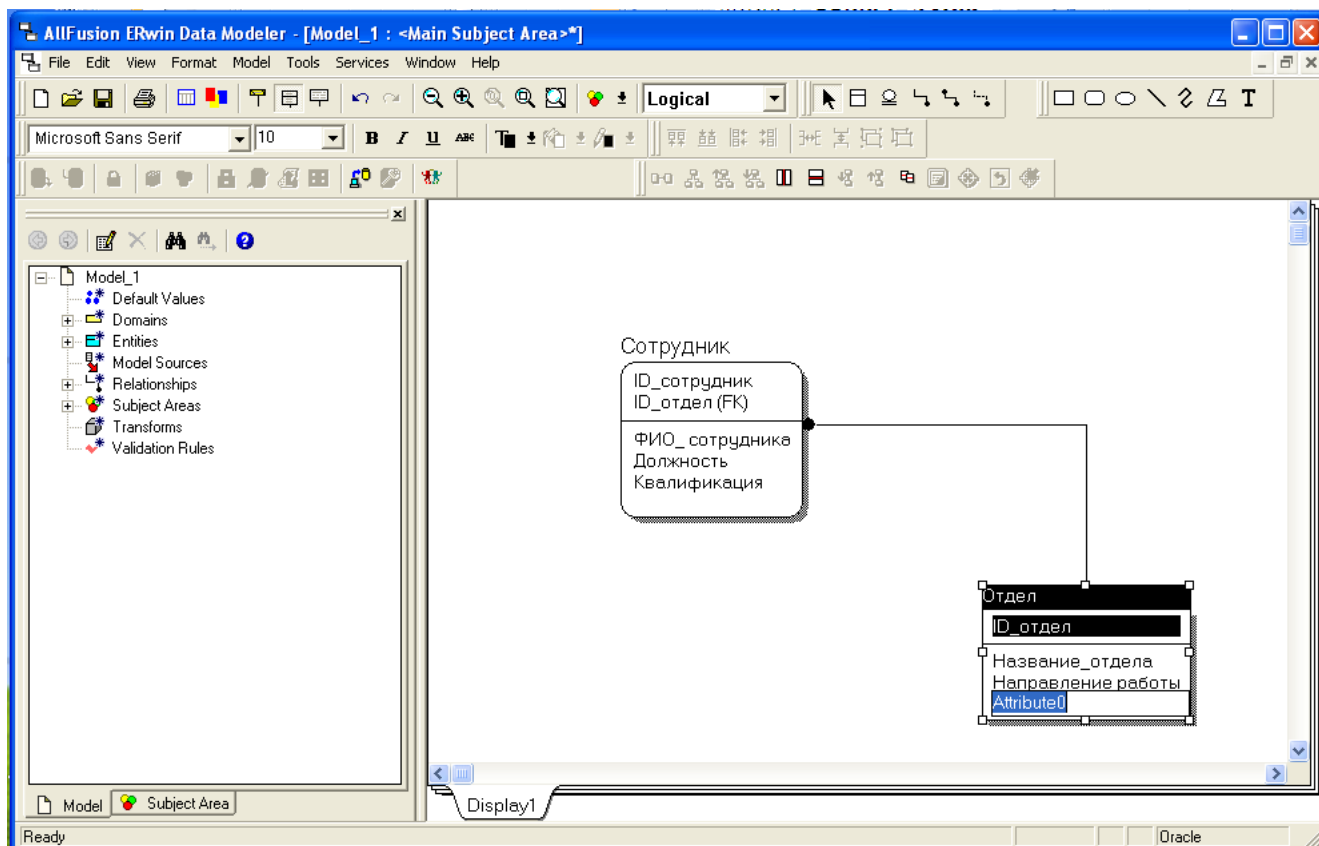


Рис. 8 Создание модели базы данных

Как было упомянуто выше, ERwin предлагает несколько форм отображения одной и той же модели. Говорят, что модель рассматривается на различных уровнях отображения.

ERwin предлагает несколько режимов просмотра изображения. Эти уровни и режимы бывают полезны при демонстрации разных типов информации на разных стадиях построения модели и при передаче информации о модели разным аудиториям. Эти уровни задаются в меню *Display Level* контекстного меню диаграммы (рис. 1.9).

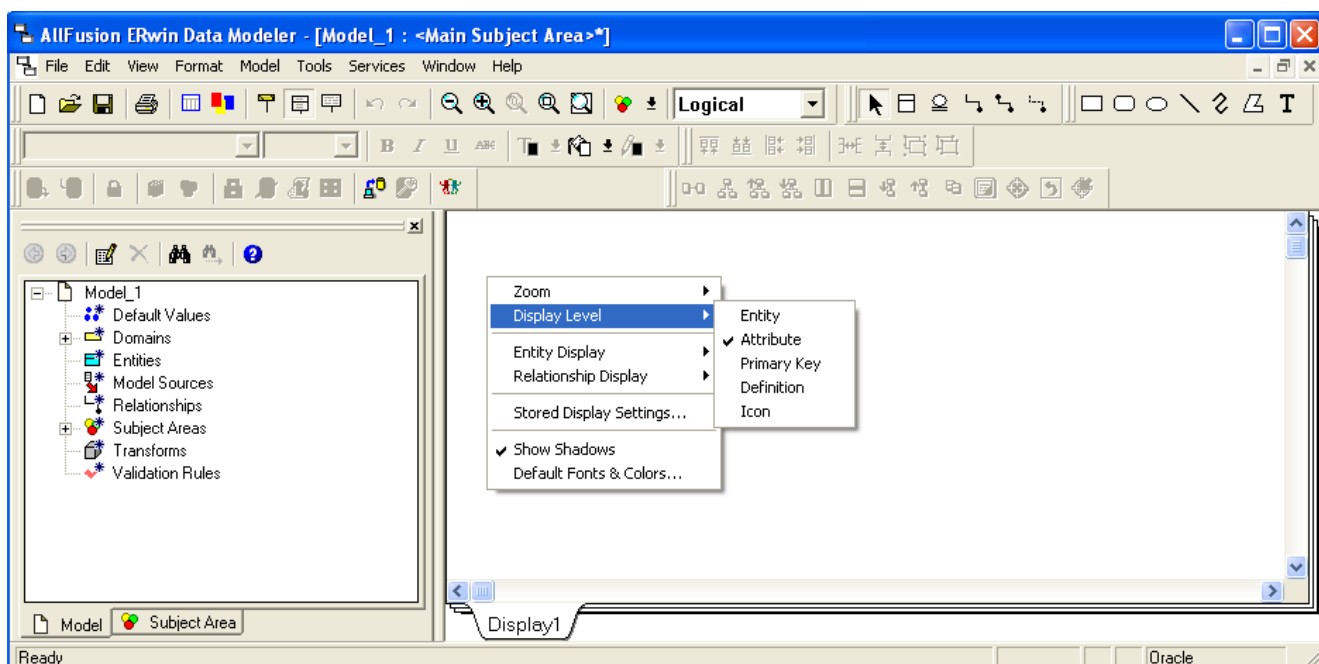


Рис. 9. Окно выбора уровня отображения элементов диаграммы

Уровень демонстрации сущности (Entity Display Level). На уровне сущности имя каждой сущности выводится на экран, заключенное в окно сущности. Никакая другая информация о сущностях на экране не появляется. Этот уровень хорошо использовать, когда только планируется модель.

Уровень демонстрации определений сущности (Entity Definition Display Level). На этом уровне размер окна сущности увеличивается и в него включается определение сущности.

Уровень демонстрации атрибутов (Attribute Display Level). На уровне демонстрации атрибутов каждая сущность в диаграмме открывается, показываются ее атрибуты, причем атрибуты, которые являются первичными ключами, расположены над чертой, а неключевые атрибуты - под чертой в окне сущности.

Уровень демонстрации первичных ключей (Primary Key Display Level). На этом уровне демонстрируются только те атрибуты, которые являются первичными ключами (те, которые находятся над чертой в окне сущности) каждой сущности. Атрибуты, которые находятся под чертой, не выводятся на экран, но они будут выведены снова при переходе обратно на уровень атрибутов.

Уровень пиктограммы (Icon Level). Для презентации можно использовать возможность, имеющуюся в ERwin: помещение изображения типа bitmap в окно сущности. Для пиктограммы сущности должен использоваться файл Windows bitmap (расширение .BMP). При желании можно для каждой сущности задать свой bitmap.

Практическое задание

1. Внимательно прочитать теоретическое введение и разобраться в структуре пользовательского интерфейса ERwin.

2. Согласно описанному методу разработать логическую модель следующей базы данных. Пусть имеется агентство по ведению статистики гражданского статуса граждан. Картотека этого агентства содержит перечень мужчин и женщин отдельного города, где для каждого мужчины или женщины известны их фамилия, имя, отчество, дата рождения, адрес места жительства и краткая характеристика. Кроме этого, имеется вторая картотека, в которой перечислены все браки между гражданами (мужчина и женщина могут быть женаты только раз). Необходимо разработать такую модель базы данных, которая позволяла бы сохранить всю эту информацию. При решении задачи предлагается опираться на сущности «мужчина» и «женщина».

2.1. Создать новую диаграмму.

2.2. Разместить на диаграмме сущности «мужчина» и «женщина»

2.3. Добавить в созданные сущности атрибуты согласно условию (использовать домен *string*)

2.4. Добавить в диаграмму не идентифицирующее отношение «мужчина женат на женщине»

2.5. Сохранить полученную модель.

3. Дополнить построенную модель так, чтобы она позволяла определить всех детей каждой женщины (если они у нее есть). Полученную модель сохраните.

5.3. Тематика письменных работ обучающихся

В течение изучения дисциплины «Проектирование информационных систем» обучающиеся должны выполнить самостоятельную работу по одной из предложенных ниже тем:

1. Разработка функциональной модели бизнес-процесса «Проектирование и реализация информационной системы».
2. Разработка функциональной модели бизнес-процесса «Создание компьютерной сети».
3. Разработка функциональной модели бизнес-процесса «Организация участка по обслуживанию и ремонту оргтехники на предприятии».
4. Разработка функциональной модели бизнес-процесса «Внедрение информационной системы».

5. Разработка функциональной модели бизнес-процесса «Покупка товаров в Интернет-магазине».
6. Разработка функциональной модели бизнес-процесса «Тестирование сайта».
7. Разработка функциональной модели бизнес-процесса «Проектирование базы данных».
8. Разработка функциональной модели бизнес-процесса «Внедрение системы автоматизированного проектирования».
9. Разработка функциональной модели бизнес-процесса «Проведение предпроектного обследования подразделения организации (предприятия)».
10. Разработка функциональной модели бизнес-процесса «Организация повышения квалификации сотрудников предприятия (организации)».
11. Разработка функциональной модели бизнес-процесса предоставления государственной или муниципальной услуги на базе многофункционального центра.
12. Разработка функциональной модели бизнес-процесса «Прием клиента».
13. Разработка функциональной модели бизнес-процесса «Оказание услуги клиенту».
14. Разработка функциональной модели бизнес-процесса «Разработка веб-сайта для предприятия (организации)».
15. Разработка функциональной модели бизнес-процесса «Предоставление информационных услуг с помощью веб-сайта предприятия (организации)».
16. Разработка функциональной модели бизнес-процесса «Учет кадров».
17. Разработка функциональной модели бизнес-процесса «Учет товарно-материальных ценностей на складе».
18. Разработка функциональной модели бизнес-процесса «Учет передачи материалов в производство».
19. Разработка функциональной модели бизнес-процесса «Оказание услуг в туристском агентстве».
20. Разработка функциональной модели бизнес-процесса «Кредитование физических лиц».
21. Разработка функциональной модели бизнес-процесса «Кредитование юридических лиц».
22. Разработка функциональной модели бизнес-процесса «Разработка плана развития предприятия».
23. Разработка функциональной модели бизнес-процесса «Учет договоров в коммерческой организации».
24. Разработка функциональной модели бизнес-процесса «Закупка материалов у поставщика».
25. Разработка функциональной модели бизнес-процесса «Учет горюче-смазочных материалов в коммерческой компании».
26. Разработка функциональной модели бизнес-процесса «Учет кассовых операций».
27. Разработать функциональную модель бизнес-процесса «Заключение договоров в коммерческой компании».
28. Разработать функциональную модель бизнес-процесса «Учет заказов в коммерческой организации».
29. Разработка функциональной модели бизнес-процесса «Деятельность отдела продаж коммерческой компании».
30. Разработка функциональной модели бизнес-процесса «Деятельность страхового агента».
31. Функциональное моделирование бизнес-процесса «Продвижение в сети веб-сайта предприятия (организации)».
32. Функциональное моделирование бизнес-процесса «Работа интернет-магазина».
33. Разработка функциональной модели начального обучения пользователей

5.4. Перечень вопросов промежуточной аттестации по дисциплине

Вопросы к зачету:

1. Подходы к построению и проектированию информационных систем.
2. Принципы системного подхода к созданию ИС.
3. Понятие технологии и операции проектирования. Требования к технологии проектирования.
4. Классификация технологий проектирования ИС. Выбор технологии проектирования ИС.
5. Понятие метода проектирования ИС, их классификация.

6. Классификация средств проектирования ИС.
7. Основные стадии жизненного цикла проектирования ИС. Модели жизненного цикла ИС.
8. Классификация стандартов на проектирование и разработку информационных систем.
9. Международный стандарт ISO/IEC 12207: 1995-08-01.
10. Стандарты комплекса ГОСТ34.
11. Методика Oracle CDM.
12. Методологии моделирования предметной области. Классификация методологий. Сравнительный анализ.
13. DFD – технология: составляющие. Диаграммы потоков данных: объекты диаграмм.
14. Диаграммы потоков данных: словари данных, спецификации процессов.
15. Диаграммы «сущность - связь». Сущности, отношения и связи. Атрибуты, категоризация сущностей.
16. Диаграммы переходов состояний. Назначение, объекты, правила и способы построения.
17. Метод моделирования процессов IDEF3. Основные объекты.
18. Метод функционального моделирования SADT (IDEF0). Основные объекты. Типы взаимосвязей между блоками.
19. Последовательность создания моделей IDEF0.
20. Понятие канонического проектирования, его особенности. Стадии и этапы процесса проектирования ИС.
21. Особенности типового проектирования. Понятие типового элемента. Классификация и примеры типовых информационных систем.
22. Состав работ на предпроектной стадии, на стадии технического и рабочего проектирования.
23. Состав работ на стадии ввода в действие ИС, эксплуатации и сопровождения.
24. Проектирование пользовательского интерфейса.
25. Понятие CASE-технологии проектирования ИС. Основные принципы Case-технологии. Факторы эффективности Case-технологии.
26. Классификация CASE-средств, стратегия их выбора.
27. Документация проекта по созданию информационной системы: Техническое задание (ТЗ), ЧТЗ, РП и т.д.
28. Назначение технико-экономического обоснования, его основные компоненты.

Раздел 6. Оценочные средства промежуточной аттестации (с ключами)

1. К какому виду обеспечения относится следующее утверждение: «Эта обеспечивающая подсистема представляет собой комплекс нормативных документов, регламентирующих деятельность персонала при функционировании ИС»

- а) правовое
- б) информационное
- в) организационное
- г) эргономическое

Правильный ответ: в)

1. Совокупность пошаговых процедур, определяющих последовательность технологических операций проектирования; критериев и правил, используемых для оценки результатов выполнения технологических операций; графических и текстовых средств, используемых для описания проектируемой системы

- а) технологии проектирования
- б) методологии проектирования
- в) жизненного цикла
- г) стандарта проектирования

Правильный ответ: а)

2. Что является элементом технологического процесса проектирования ИС

- а) технологические карты
- б) стандарты проектирования
- в) технологическое оборудование
- г) технологическая операция

Правильный ответ: г)

3. Связь в модели ER

- а) глаголы
- б) существительное
- в) прилагательное
- г) союз

Правильный ответ: а)

4. Программно-аппаратный комплекс, предназначенный для автоматизированного сбора, хранения, обработки и выдачи информации называют _____

Правильный ответ: информационная система

5. Какие основные понятия используются при создании диаграммы потоков данных?

- а) потоки данных
- б) процессы преобразования входных потоков данных в выходные
- в) внешние источники и получатели данных
- г) хранилища, требуемые процессам для своих операций
- д) операционные системы

Правильный ответ: а), б), в), г)

6. Какие основные понятия используются при создании функциональной диаграммы предметной области

- а) функциональный блок

- б) интерфейсная дуга
- в) база данных
- г) операционная система
- д) декомпозиция

Правильный ответ: а), б), д)

7. Указание множества допустимых информационных конструкций, операций над данными и множества ограничений для хранимых значений, - это _____ данных

Правильный ответ: модель

8. Формулирование требований к информационной системе – это _____ стадия проектирования ИС

Правильный ответ: предпроектная

9. В проектировании ИС модель, основанная на логике предикатов или правилах, которые описывают знания в форме "если - то", называется _____

Правильный ответ: продукционная модель

10. Непрерывный процесс, который начинается с момента принятия решения о создании ИС и заканчивается в момент полного изъятия системы из эксплуатации называется _____ информационной системы.

Правильный ответ: жизненный цикл

11. Компьютерная программа, преобразующие опыт ведущих специалистов в какой-либо области знаний в форму эвристических правил на основе использования технологии искусственного интеллекта, называется _____

Правильный ответ: экспертная система

12. Укажите прямую последовательность этапов жизненного цикла ИС

- а) удаление
- б) проектирование
- в) внедрение
- г) реализация
- д) эксплуатация
- е) планирование и анализ требований

Правильный ответ: е) - б) - г) - в) - д) - а)

13. Расположите следующие единицы информации в порядке возрастания синтаксической сложности

- а) показатель
- б) реквизит
- в) база данных
- г) составная единица информации (СЕИ)

Правильный ответ: б) - а) – г) – в)

14. Установите соответствие

1) Ассимилированная информация	а) сведения, рассматриваемые в момент передачи информации от источника к приемнику
--------------------------------	--

2) Документированная информация	б) представление сообщений в сознании человека, наложенное на систему его понятий и оценок
3) Передаваемая информация	в) сведения, зафиксированные в знаковой форме на каком-то физическом носителе

Правильный ответ:

- 1) – б)
- 2) – в)
- 3) – а)

15. Установите соответствие основных свойств ИС и их определений

1) Свойство относительности	а) устанавливает, что состав элементов, взаимосвязей, входов, выходов, целей и ограничений ЭИС зависит от целей применения
2) Делимость	б) указывает на согласованность цели функционирования всей ЭИС с целями функционирования ее подсистем и элементов
3) Свойство целостности	в) означает, что ЭИС можно представить состоящей из относительно самостоятельных частей - подсистем, каждая из которых может рассматриваться как система

Правильный ответ:

- 1) – б)
- 2) – в)
- 3) – а)

16. Установите соответствие терминов и их определений, касающихся подсистем обеспечения ИС

1) Техническое обеспечение	а) совокупность правовых норм, определяющих создание, юридический статус и функционирование ЭИС
2) Информационное обеспечение	б) комплекс аппаратных средств- компьютеры, устройства сбора, обработки и передачи информации, линии связи, оргтехника
3) Программное обеспечение	в) совокупность системы классификации и кодирования информации, схем информационных потоков и методологии построения баз данных
4) Организационное обеспечение	г) совокупность алгоритмов и программ для реализации целей информационной системы и нормального функционирования технических средств
5) Правовое обеспечение	д) совокупность методов и средств, регламентирующих взаимодействие работников между собой и с техническими средствами в процессе эксплуатации ЭИС

Правильный ответ:

- 1) – б)
- 2) – в)
- 3) – г)
- 4) – д)
- 5) – а)

17. Установите соответствие аббревиатур и названий методик графического анализа информационных систем

1) SADT	а) диаграммы потоков данных
2) DFD	б) технология структурного анализа и проектирования
3) ERD	в) диаграммы переходов состояний
4) STD	г) диаграммы «сущность-отношение»

Правильный ответ:

- 1) – б)
- 2) – а)
- 3) – г)
- 4) – в)

Раздел 7. Перечень учебной литературы, необходимой для освоения дисциплины

7.1. Основная литература

1. Тимофеев, А. В. Проектирование и разработка информационных систем : учебное пособие для СПО / А. В. Тимофеев, З. Ф. Камальдинова, Н. С. Агафонова. — Саратов : Профобразование, 2022. — 91 с. — ISBN 978-5-4488-1416-7. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/116285.html>.

2. Логанов, С. В. Объектно-ориентированные принципы разработки информационных систем : учебное пособие / С. В. Логанов, С. Л. Моругин. — Москва : Ай Пи Ар Медиа, 2022. — 217 с. — ISBN 978-5-4497-1576-0. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/118880.html>

7.2. Дополнительная литература

3. Бабич, А. В. Введение в UML : учебное пособие / А. В. Бабич. — 4-е изд. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2022. — 198 с. — ISBN 978-5-4497-1637-8. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/120473.html>

4. Васильев, Р. Б. Управление развитием информационных систем : учебник / Р. Б. Васильев, Г. Н. Калянов, Г. А. Левочкина. — 4-е изд. — Москва : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2022. — 507 с. — ISBN 978-5-4497-1654-5. — Текст : электронный // Цифровой образовательный ресурс IPR SMART : [сайт]. — URL: <https://www.iprbookshop.ru/120490.html>

7.3. Другие источники информации и средства обеспечения освоения дисциплины

5. Журнал «Бизнес. Образование. Право. Вестник Волгоградского института бизнеса» [Электронный ресурс] // URL: <http://vestnik.volbi.ru/>

6. Издательство “Открытые системы” [Электронный ресурс] // URL: <http://www.osp.ru>

7. Информационный сайт Цитфорум [Электронный ресурс] // URL: <http://www.citforum.ru>

8. ПО для организации конференций

9. СПС «КонсультантПлюс», URL: <http://www.consultant.ru/>

10. СПС «ГАРАНТ», URL: <http://base.garant.ru/>

11. Управление проектами [Электронный ресурс] // URL: <http://www.pmtoday.ru>

Раздел 8. Материально-техническая база и информационные технологии

Материально-техническая база, необходимая для осуществления образовательного процесса по дисциплине:

Материально-техническое обеспечение дисциплины включает в себя учебные аудитории для проведения занятий, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации, самостоятельной работы обучающихся.

Учебные аудитории укомплектованы специализированной мебелью и техническими средствами обучения. Помещения для самостоятельной работы обучающихся оснащены компьютерной техникой с возможностью подключения к сети Интернет.

Дисциплина может реализовываться с применением дистанционных технологий обучения. Специфика реализации дисциплины с применением дистанционных технологий обучения устанавливается дополнением к рабочей программе. В части не противоречащей специфике, изложенной в дополнении к программе, применяется настоящая рабочая программа.

Материально-техническая база, необходимая для осуществления образовательного процесса по дисциплине с применением дистанционных образовательных технологий включает в себя:

Компьютерная техника, расположенная в учебном корпусе Института (ул.Качинцев, 63, кабинет Центра дистанционного обучения):

1. Intel i 3 3.4Ghz\O3Y 4Gb\500GB\RadeonHD5450

2. Intel PENTIUM 2.9GHz\O3Y 4GB\500GB

3 личные электронные устройства (компьютеры, ноутбуки, планшеты и иное), а также средства связи преподавателей и студентов.

Информационные технологии, необходимые для осуществления образовательного процесса по дисциплине с применением дистанционных образовательных технологий включают в себя:

- система дистанционного обучения (СДО) (Learning Management System) (LMS) Moodle (Modular Object-Oriented Dynamic Learning Environment);

- электронная почта;

- система компьютерного тестирования;

- Цифровой образовательный ресурс IPR SMART;

- система интернет-связи skype;

- телефонная связь;

- Программное обеспечение для организации конференции

Обучение обучающихся инвалидов и обучающихся с ограниченными возможностями здоровья осуществляется посредством применения специальных технических средств в зависимости от вида нозологии.

При проведении учебных занятий по дисциплине используются мультимедийные комплексы, электронные учебники и учебные пособия, адаптированные к ограничениям здоровья обучающихся.

Лекционные аудитории оборудованы мультимедийными кафедрами, подключенными к звуковым колонкам, позволяющими усилить звук для категории слабослышащих обучающихся, а также проекционными экранами, которые увеличивают изображение в несколько раз и позволяют воспринимать учебную информацию обучающимся с нарушениями зрения.

При обучении лиц с нарушениями слуха используется усилитель слуха для слабослышащих людей Cyber Ear модель НАР-40, помогающий обучаемым лучше воспринимать учебную информацию.

Обучающиеся с ограниченными возможностями здоровья, обеспечены печатными и электронными образовательными ресурсами (программы, учебники, учебные пособия, материалы для самостоятельной работы и т.д.) в формах, адаптированных к ограничениям их здоровья и восприятия информации:

для лиц с нарушениями зрения:

- в форме электронного документа;

- в форме аудиофайла;

для лиц с нарушениями слуха:

- в печатной форме;

- в форме электронного документа;
- для лиц с нарушениями опорно-двигательного аппарата:**
- в печатной форме;
- в форме электронного документа;
- в форме аудиофайла.

Раздел 9. Методические указания для обучающихся по освоению дисциплины

Для успешного усвоения материала дисциплины требуются значительное время, концентрация внимания и усилия: посещение лекционных занятий и конспектирование преподаваемого материала, работа с ним дома, самостоятельная проработка материала рекомендуемых учебников и учебных пособий при самостоятельной подготовке. Особое внимание следует обратить на выполнение практических работ, практических задач по СРО, тестовых вопросов.

При самостоятельной работе с учебниками и учебными пособиями полезно иметь под рукой справочную литературу (энциклопедии) или доступ к сети Интернет, так как могут встречаться новые термины, понятия, которые раньше обучающиеся не знали.

Цель практических занятий по дисциплине «Проектирование информационных систем» - закрепление знаний по определенной теме, приобретенных в результате прослушивания лекций, получения консультаций и самостоятельного изучения различных источников литературы. При выполнении данных работ обучающиеся должны будут глубоко изучить методы и средства проектирования ИС. Получить навыки использования CASE-систем.

Перед практическим занятием обучающийся должен детально изучить теоретические материалы вопросов практики в учебниках, конспектах лекций, периодических журналах и прочее. Если при выполнении практического задания у обучающегося остаются неясности, то ему необходимо оперативно обратиться к преподавателю за уточнением.

После выполнения практического задания обучающиеся должны выполнить самостоятельную работу. Самостоятельная работа включает в себя индивидуальное задание по пройденной теме. Таким образом, каждый обучающийся выполняет только свой вариант задания. Решение практических заданий сопровождается выполнением письменного отчета в тетради. Отчет должен выполняться аккуратно, быть легко читаемым подчерком, при этом допускаются общепринятые сокращения.

При дистанционном выполнении практических работ обучающийся может самостоятельно приобрести операционные системы Windows XP, Windows 7, Windows 8, Windows 10. Программа Ramus распространяется свободно, её можно скачать с соответствующего сайта. Ответственность за установку и настройку программного обеспечения в данном случае ложится на обучающегося. Следует воспользоваться методическими указаниями по установке данных программных систем.

Результаты выполненных заданий оцениваются с учетом теоретических знаний по соответствующим разделам дисциплины, техники выполнения работы, объективности и обоснованности принимаемых решений в процессе работы с данными, качества оформления. Переход к выполнению следующего практического задания допускается только после отчета выполненной работы.

Учебно-методическое издание

Рабочая программа учебной дисциплины

Проектирование информационных систем

(Наименование дисциплины в соответствии с учебным планом)

Филиппов Михаил Владимирович

(Фамилия, Имя, Отчество составителя)
