

В. В. Бахтизин, Л. А. Глухова, С. Н. Неборский

***РАБОТА С ТРЕБОВАНИЯМИ  
В СРЕДЕ DOORS***

*Рекомендовано Учебно-методическим объединением  
вузов Республики Беларусь по образованию  
в области информатики и радиоэлектроники  
в качестве учебно-методического пособия  
для студентов учреждений, обеспечивающих получение  
высшего образования по специальности  
«Программное обеспечение информационных технологий»*

Минск БГУИР 2007

УДК 681.3 (075.8)  
ББК 32.973.202-018.2 я 7  
Б 30

Рецензенты:

зав. кафедрой программного обеспечения сетей телекоммуникаций  
Высшего государственного колледжа связи,  
д-р техн. наук, проф. А. А. Прихожий;

зав. кафедрой информатики Минского государственного  
высшего радиотехнического колледжа,  
канд. техн. наук, доц. Ю. А. Скудняков

**Бахтизин, В. В.**

Б30 Работа с требованиями в среде DOORS : учебно-метод. пособие /  
В. В. Бахтизин, Л. А. Глухова, С. Н. Неборский. – Минск : БГУИР, 2007. –  
40 с. : ил.

ISBN 978-985-488-195-9

Пособие посвящено одной из тем дисциплины «Технологии разработки программного обеспечения». Описаны правила и процесс работы с требованиями к программным средствам. Рассмотрена инструментальная среда Telelogic DOORS 8.1. На конкретном примере пояснена разработка требований в системе DOORS. Приведены общие требования по выполнению лабораторной работы и варианты индивидуальных заданий.

УДК 681.3 (075.8)  
ББК 32.973.202-018.2 я 7

ISBN 978-985-488-195-9

© Бахтизин В. В., Глухова Л. А.,  
Неборский С. Н., 2007  
© УО «Белорусский государственный  
университет информатики  
и радиоэлектроники», 2007

# СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	4
1. РАЗРАБОТКА И АНАЛИЗ ТРЕБОВАНИЙ, УПРАВЛЕНИЕ ТРЕБОВАНИЯМИ .....	4
1.1. Основы инженерии требований .....	4
1.2. Процесс работы с требованиями.....	5
1.3. Методы моделирования требований.....	7
2. СИСТЕМА УПРАВЛЕНИЯ ТРЕБОВАНИЯМИ DOORS 8.1 .....	8
2.1. Общие сведения о системе DOORS .....	8
2.2. Основы DOORS.....	9
2.2.1. DOORS как многопользовательская система.....	9
2.2.2. Папки, проекты и модули DOORS .....	9
2.2.3. Формальные модули.....	13
2.3. Работа с требованиями .....	16
2.3.1. Формальные модули как контейнеры объектов.....	16
2.3.2. Атрибуты.....	17
2.3.3. Колонки и виды в формальном модуле .....	19
2.3.4. Поиск, фильтрация и сортировка.....	20
2.3.5. Связи между требованиями .....	24
2.3.6. Подозрительные связи.....	27
2.3.7. Модули связей .....	28
2.3.8. Средства анализа связей.....	30
2.4. Управление изменениями требований.....	34
2.4.1. Инструменты управления изменениями требований .....	34
2.4.2. Базовые версии .....	34
2.4.3. Цифровая подпись .....	35
2.4.4. Система предложений о внесении изменений .....	36
3. ПРИМЕР ЗАДАНИЯ ТРЕБОВАНИЙ В СИСТЕМЕ DOORS .....	37
3.1. Постановка задачи .....	37
3.2. Определение требований.....	37
4. КОНТРОЛЬНЫЕ ВОПРОСЫ .....	38
5. ОБЩИЕ ТРЕБОВАНИЯ К ЛАБОРАТОРНОЙ РАБОТЕ .....	38
5.1. Порядок выполнения работы .....	38
5.2. Возможные последовательности шагов при выполнении работы в среде DOORS 8.1.....	39
5.2.1. Работа с объектами требований .....	39
5.2.2. Использование функций поиска, фильтрации, сортировки .....	39
5.2.3. Работа со связями требований .....	40
5.3. Варианты индивидуальных заданий .....	40
ЛИТЕРАТУРА .....	41

## ВВЕДЕНИЕ

Разработка любого программного средства (ПС) начинается с определения предъявляемых к нему требований. От того насколько правильно, полно и точно сформулированы требования, непосредственно зависит качество конечного программного продукта. Требования можно рассматривать как свойства, которыми должен обладать этот продукт. Очевидно, чем сложнее разрабатываемое ПС, тем более сложную структуру имеют требования. Следовательно, для повышения качества самих требований необходимы специализированные ПС. Такими ПС являются системы управления требованиями (СУТ).

Выбор СУТ – это важный этап процесса разработки ПС, ведь, по сути, механизм требований – это единственно возможный способ взаимодействия заказчика и разработчика. Сегодняшний рынок систем управления требованиями достаточно многообразен. Так, можно назвать такие СУТ, как Xarware ActiveFocus, Borland CaliberRM, Rational RequisitePro, Telelogic DOORS. Последнее ПС (СУТ DOORS) по праву является одним из лидеров в области СУТ.

Telelogic DOORS предоставляет следующие возможности: формулирование требований; определение всей необходимой для управления требованиями информации; контроль состояния требований; гибкая политика изменения требований; проверка соответствия всей информации, описывающей требования, установленным стандартам и правилам; обеспечение полного анализа влияния изменения требований.

Начальные концепции DOORS были разработаны доктором Ричардом Стивенсом (Richard Stevens) в 1993 г. DOORS много раз признавалась агентствами статистических исследований лучшей системой управления требованиями (Yphise, Standish Group, Software Development).

Система DOORS активно развивается. В настоящее время DOORS – это мощная функциональная система. В пособии рассматривается одна из последних версий DOORS – Telelogic DOORS 8.1.

## 1. РАЗРАБОТКА И АНАЛИЗ ТРЕБОВАНИЙ, УПРАВЛЕНИЕ ТРЕБОВАНИЯМИ

### 1.1. Основы инженерии требований

Ключевым понятием в инженерии требований является понятие *требования*. Требования можно рассматривать как свойства, которыми должен обладать программный продукт в соответствии с пожеланиями пользователя [4]. Посредством требований формализуются задачи, которые призвано решать создаваемое ПС. Именно поэтому требования представляют собой универсальный язык общения тех, кто реализует требование (разработчиков), и тех, кто нуждается в этой реализации (заказчиков).

В инженерии требований принято выделять следующие *роли работающих с требованиями*: пользователи, эксперты предметной области, менеджеры проекта, системные аналитики, программисты, инженеры по тестированию, инженеры по установке и настройке ПС (системы). Таким образом, требования представляют собой механизм взаимодействия сторон. Количество этих сторон достаточно велико. Поэтому выделяют различные *виды (уровни) требований*.

*Бизнес-требования* содержат высокоуровневые цели организации или заказчиков ПС (системы). *Требования пользователей* описывают цели и задачи, которые пользователям позволит решить ПС (система). К способам представления этого вида требований относятся варианты и сценарии использования. *Системные требования* описывают требования к системе, которая в общем случае состоит из аппаратных средств, программных средств и ручных операций. Из системных требований выделяется уровень *требований к программным средствам*. Данные требования представляют собой требования к программным компонентам системы и в свою очередь могут быть детализированы на несколько уровней.

По назначению требования можно разделить на функциональные и нефункциональные. *Функциональные требования* определяют функции ПС (системы), позволяющие пользователям выполнить свои задачи в рамках бизнес-требований. *Нефункциональные требования* описывают другие свойства ПС (системы), например, требуемую производительность, требования к качеству, окружающую среду.

Формирование требований является начальным этапом любой технологии разработки ПС. Требования могут разрабатываться однократно (при каскадной стратегии разработки) или итерационно (при эволюционной или инкрементной стратегии).

## 1.2. Процесс работы с требованиями

Процесс работы с требованиями состоит из следующих *задач* [6]:

1. Выявление требований.
2. Анализ и согласование требований.
3. Документирование требований.
4. Аттестация требований.
5. Управление требованиями.

**1. Выявление требований.** Выявление требований является базовой задачей процесса работы с требованиями. При выявлении требований важную роль играет понимание предметной области, проблемы, технологических процессов, потребностей и ограничений заинтересованными в ПС (системе) лицами.

Основными *техниками выявления требований* являются [6]:

– интервьюирование (техника, при которой пользователи ПС или системы отвечают на вопросы, тем самым формируя свои потребности и цели);

- мозговой штурм (пользователи ПС или системы в процессе живого и целенаправленного обсуждения формулируют свои требования);
- сценарии использования ПС или системы (требования формируются на основе того, как ПС или система будет использована);
- наблюдение (требования фиксируются путем наблюдения за действиями пользователя);
- повторное использование требований (требования, описывающие некоторые свойства в уже реализованном ПС или системе, могут быть перенесены на разрабатываемое ПС или систему);
- ролевые игры (разработчики выясняют требования пользователей путем анализа проблемы с точки зрения пользователя);
- прототипирование требований (быстрая реализация непонятого или недостаточно четко сформулированного требования с тем, чтобы определить, того ли результата ожидает пользователь).

**2. Анализ и согласование требований.** Анализ требований позволяет довести качество требований до надлежащего уровня. Результатом анализа является уточнение, конкретизация, разбиение или удаление требований. Результаты анализа требований согласуются между разработчиком и заказчиком.

Анализ требований привязан к определенному контексту. Такой контекст формируется на основе связей требований. Связанными могут быть как требования разных уровней, так и требования одного уровня (табл. 1.1).

Таблица 1.1

Виды анализа связей требований

Вид анализа связей требований	Тип анализируемых связей	Главная цель
Анализ влияния	Исходящие связи	Установить, на что влияет изменение рассматриваемого требования
Анализ последствий (анализ трассируемости)	Входящие связи	Установить, что потребуется для изменения рассматриваемого требования
Анализ покрытия	Входящие и исходящие связи	Определить, все ли учтено при определении требований и их связей, корректны ли связи требований

Анализируя требования, следует четко представлять иерархию уровней требований и то, как каждый уровень соотносится с тестом, проверяющим реализацию требования.

**3. Документирование требований.** Чтобы обеспечить связь между

пользователями ПС (системы), разработчиками, менеджерами проекта, требования должны быть зафиксированы в *спецификации требований*. При разработке спецификации требований должны быть учтены следующие *факторы*: свойства ПС или системы (функции, внешние интерфейсы, производительность, характеристики качества); окружающая среда; *свойства требований* (корректность, недвусмысленность, полнота, непротиворечивость, упорядоченность по важности и стабильность, контролируемость, модифицируемость, трассируемость); возможность совместной подготовки требований; возможности развития требований; необходимость прототипирования требований; требования к этапу проектирования ПС (или системы); общие требования к проекту разработки [1].

**4. Аттестация требований.** Целью аттестации требований является определение факта готовности требований к передаче разработчикам для реализации. Аттестация требований включает: проверку полноты требований; проверку на соответствие принятым в организации стандартам; проверку отсутствия противоречивых и конфликтующих требований; проверку отсутствия технических ошибок.

**5. Управление требованиями.** Управление требованиями – это деятельность по поддержанию требований в таком состоянии, при котором обеспечивается подчинение требований единой структуре и четкая организация взаимосвязей между ними. Управление требованиями охватывает: управление изменениями требований; управление связями между требованиями; распределение ролей для выполнения работ, связанных с управлением требованиями; постоянный контроль качества требований; контроль реализации требований.

Невозможно эффективно управлять требованиями без трассируемости требований. *Трассируемость требований* определяет зависимости и логические связи отдельных требований между собой и с другими элементами ПС (системы). К этим элементам относятся, например, бизнес-правила, архитектура и другие компоненты проекта ПС (системы), модули исходного кода, варианты тестирования, файлы справки [4]. Управление требованиями является сложным процессом, который требует для своей поддержки использования СУТ.

### **1.3. Методы моделирования требований**

С целью повышения эффективности выявления и анализа требований используются различные методы моделирования предметной области. Из них наиболее широко используются:

- методы структурного анализа (методология структурного анализа и проектирования SADT [2], диаграммы потоков данных DFD [3]);
- методы перспектив (управляемое формулирование требований CORE и определение требований, основанное на перспективах VORD [5]);

– семантические методы (информационное моделирование, ЕМТ – расширенная реляционная модель Кодда, SDM – семантическая модель данных Хаммера и Мак Леода [6]);

– объектно-ориентированные методы (методы объектно-ориентированного анализа, моделирования и проектирования [5], основой для которых является универсальный язык моделирования UML);

– формальные методы (основаны на математических подходах – Z-метод Спрея, язык венского определения VDL Джонса, язык для определения временной последовательности LOTOS Бьернера, В-метод Абриала [5]).

## 2. СИСТЕМА УПРАВЛЕНИЯ ТРЕБОВАНИЯМИ DOORS 8.1

### 2.1. Общие сведения о системе DOORS

DOORS – это многопользовательская клиент-серверная система управления требованиями. Все данные, обрабатываемые DOORS, хранятся в единой базе данных на сервере, что обеспечивает их согласованность.

Использование централизованного хранилища данных имеет много преимуществ. Например, DOORS позволяет блокировать данные, то есть использовать режим монопольного редактирования информации. Единая база данных содержит историю всех вносимых изменений, что дает возможность сравнения разных версий требований. Мощный редактор сравнения позволяет проводить различные манипуляции над разными версиями требований.

DOORS реализует гибкую политику управления пользователями и группами пользователей, предоставляя им различные права доступа к данным и объектам базы данных. DOORS поддерживает операцию репликации данных. Репликация предполагает выделение части базы данных и передачу ее пользователю. С полученными данными пользователь может работать локально. Затем данный фрагмент базы данных корректно вставляется обратно.

Для управления требованиями в DOORS существуют связи. Связи позволяют анализировать влияние изменения одних требований на другие.

Для расширения возможностей DOORS разработан *язык DXL* (DOORS eXtension Language) – скриптовый язык, который позволяет автоматизировать специфические процедуры или сложные задачи и реализовывать триггеры (функции, срабатывающие при наступлении некоторого события).

Для удаленной работы с DOORS используется компонент DOORSnet. Он позволяет получить доступ к базе данных и ко всем функциям СУТ DOORS через Интернет при помощи браузера.

Система DOORS может интегрироваться с другими программными продуктами компании Telelogic, что позволяет автоматизировать весь жизненный цикл программных средств.



## 2.2. Основы DOORS

### 2.2.1. DOORS как многопользовательская система

Каждый пользователь DOORS имеет определенный набор прав доступа. Права доступа определяют, кто может работать с конкретным объектом базы данных и что он может делать. В DOORS определены следующие *типы пользователей*: стандартный пользователь, менеджер проекта, менеджер базы данных, особый пользователь.

Информация в DOORS хранится в базе данных. Чтобы войти в систему DOORS и увидеть структуру базы данных, необходимо при загрузке приложения–клиента ввести имя пользователя и пароль (рис. 2.1). Если введенные данные верны, будет загружено главное окно DOORS (рис. 2.2). В этом окне отображается структура базы данных. Доступ к ее объектам осуществляется на основе прав доступа, которыми обладает текущий пользователь.

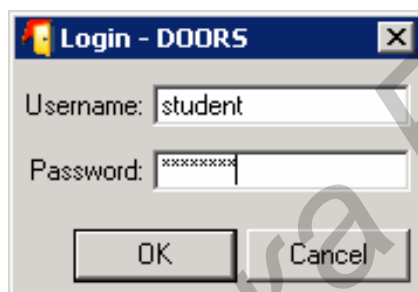


Рис. 2.1. Окно ввода имени пользователя и пароля

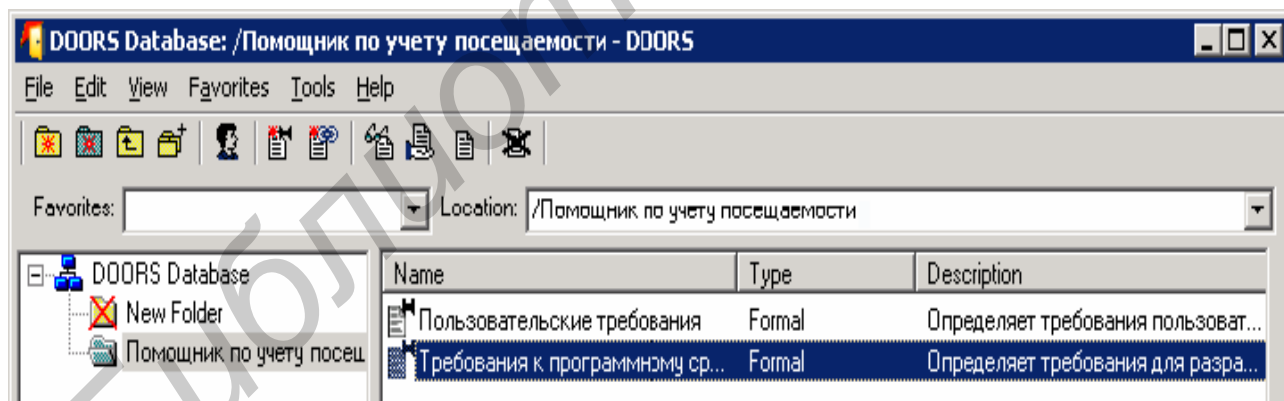
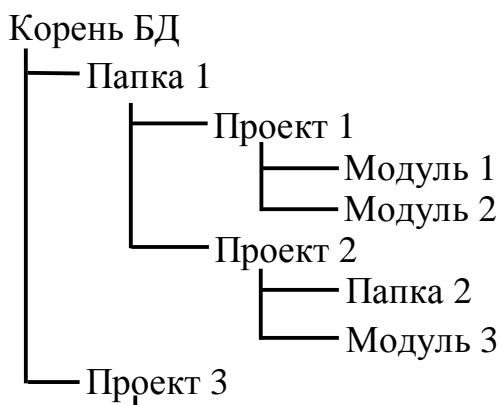


Рис. 2.2. Главное окно DOORS

### 2.2.2. Папки, проекты и модули DOORS



В главном окне приложения отображается структура базы данных (БД) либо структура проектов DOORS. Хранимая информация представляется

иерархически в виде проектов, папок и модулей. Папки и проекты используются для структуризации данных. Информация о требованиях и связях между ними находится в модулях. На корневом уровне БД можно использовать лишь папки и проекты. Пример структуры БД DOORS представлен на рис. 2.3. Проводник DOORS может отобразить данные в виде непосредственно базы данных (пункты меню View > Database View) либо в виде проектов (View > Project View). Вид непосредственно базы данных представлен на рис. 2.4, вид проектов DOORS – на рис. 2.5.

Рис. 2.3. Пример структуры базы данных DOORS

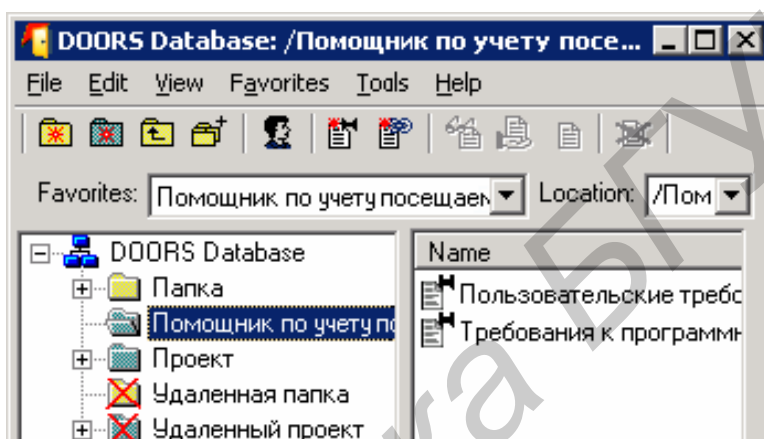


Рис. 2.4. Вид базы данных в проводнике DOORS

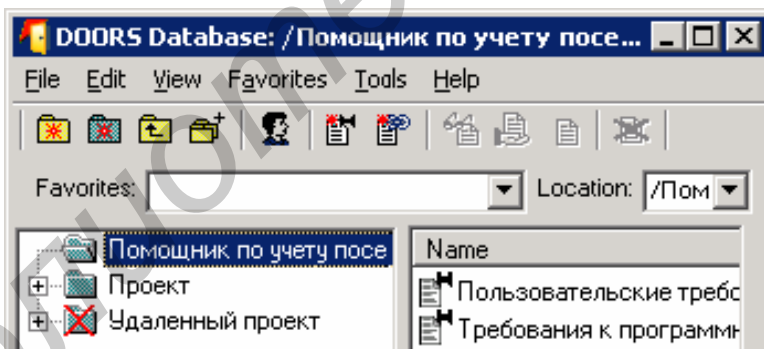


Рис. 2.5. Вид проектов в проводнике DOORS

На левой панели проводника информация отображается в иерархическом виде. Правая панель содержит объекты одного уровня. Опции меню позволяют адаптировать отображаемую информацию для конкретных целей. Например, можно задать опции проводника, при которых не будут отображаться папки, проекты или модули.

Ни один из объектов DOORS не может рассматриваться без связанного с ним контекста. Всегда важна информация о положении этого объекта в структуре базы данных. Путь к объекту базы данных записывается от корня к объекту, причем все родительские объекты отделяются символом «/».

Корректным является, например, следующий путь:  
/Папка1/Проект2/Модуль3.

Существует *три вида модулей*: формальные (содержат описания требований), модули связей (хранят наборы связей) и описательные модули (в настоящее время считаются устаревшими и на практике используются редко).

Папки используются как контейнеры статических (мало изменяющихся) данных. Проекты содержат информацию, относящуюся к конкретному проекту. Эта информация подвержена постоянным изменениям.

Главное отличие между папкой и проектом состоит в том, что проекты могут быть реплицированы (выделены в отдельную сущность из базы данных). Поэтому имена проектов должны быть уникальны в пределах всей базы данных. Имена папок должны различаться в пределах текущего уровня иерархии для родительской папки или проекта. Графически в проводнике DOORS папки отображены желтым цветом. Цвет проектов – синий.

Создавать папки, проекты, модули можно различными способами. Например, можно использовать контекстное меню проводника DOORS и его пункты New > Folder (Project, Formal Module), а также панель инструментов или главное меню (рис. 2.6).

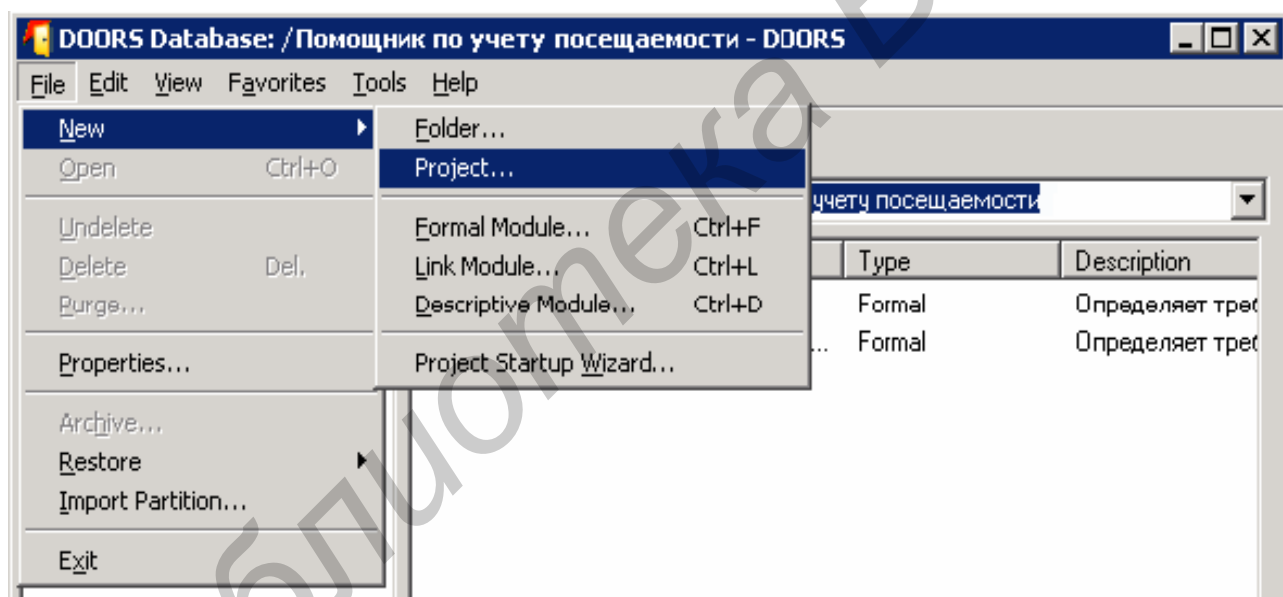


Рис. 2.6. Создание папки, проекта или модуля

Каждый объект DOORS обладает свойствами. Как правило, посредством свойств можно настроить права доступа к объекту и изменить общую информацию (например имя объекта и его описание).

При удалении папки (проекта, модуля) не происходит уничтожения находящихся в ней данных. Папка лишь помечается как удаленная, и пользователи не имеют доступа к информации этой папки. Чтобы уничтожить данные, необходимо очистить папку после ее удаления. Очистка удаленной папки приводит к полному исчезновению папки из базы данных.

Чтобы восстановить удаленную папку (проект, модуль), необходимо

вначале отобразить ее в проводнике DOORS. Для этого требуется настройка вида отображаемой информации (View > Show Deleted). Далее необходимо выделить папку, которая подлежит восстановлению, и выполнить команду восстановления (File > Undelete) в меню (рис. 2.7).

Если работа с системой DOORS заканчивается некорректно (например по причине сбоя работы компьютера), все модули, в которых проводилось редактирование, а также открытые проекты и папки блокируются. Зabloкированные модули становятся доступными только для чтения. Их редактирование невозможно. Зabloкированные папки нельзя переместить, переименовать. Чтобы разблокировать папки или модули, необходимо воспользоваться менеджером блокировок. Для его активации используется команда меню Tools > Manage Locks (рис. 2.8).

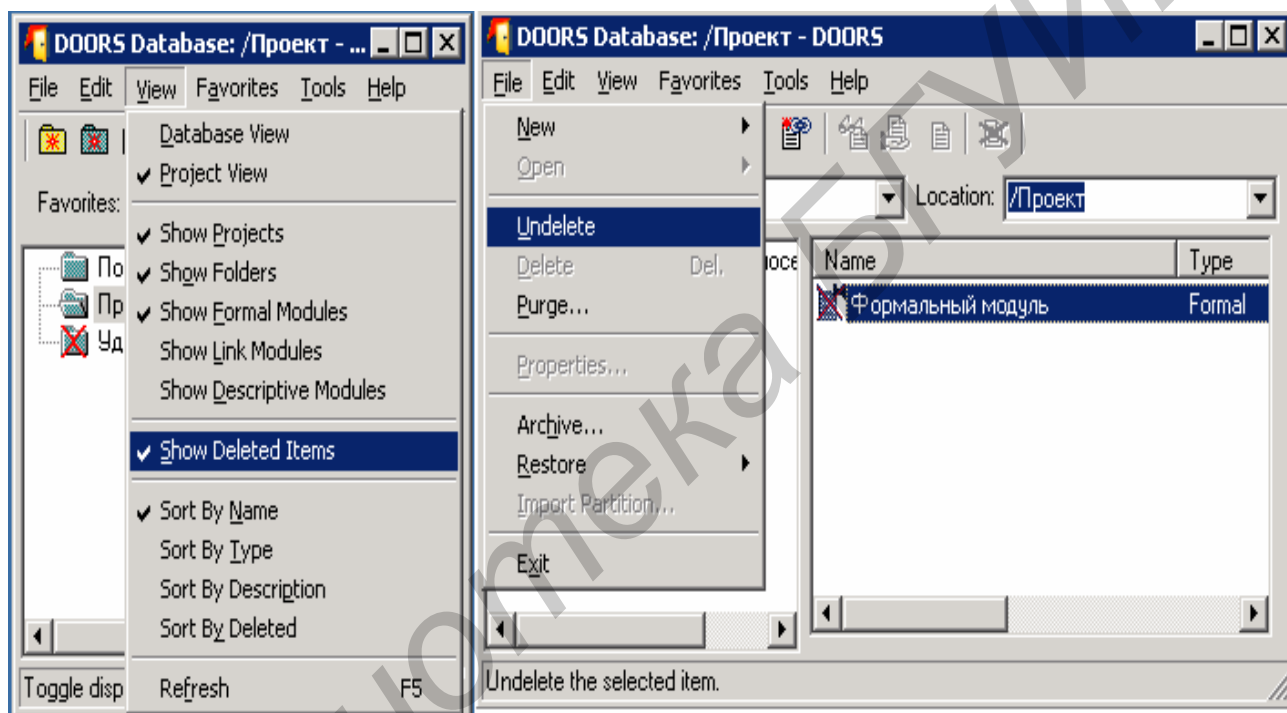


Рис. 2.7. Восстановление удаленных папок, проектов и модулей

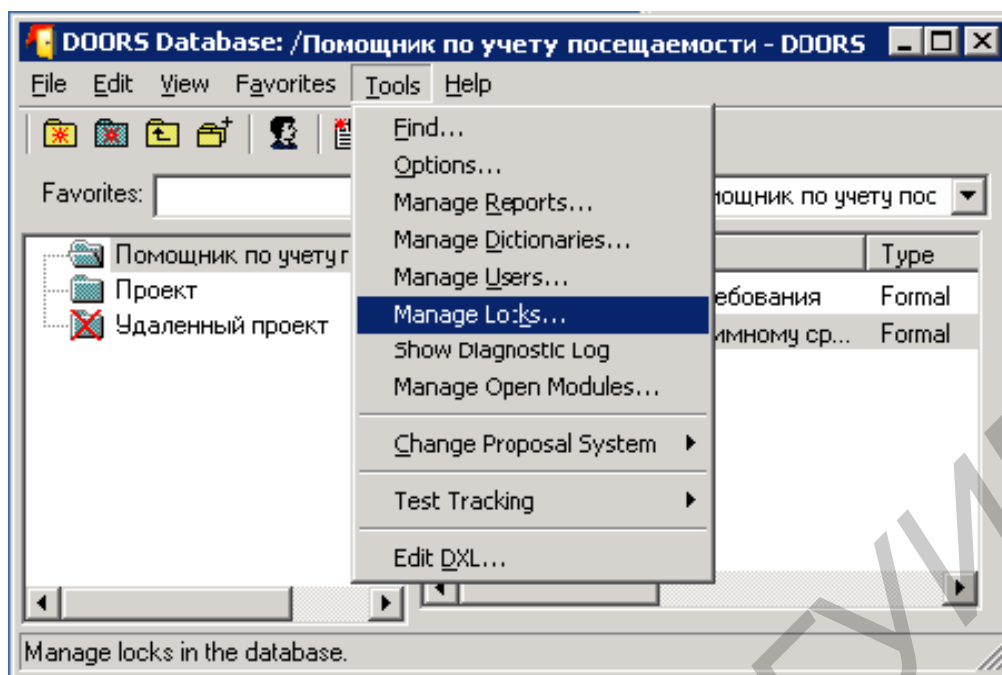


Рис. 2.8. Управление блокировками

### 2.2.3. Формальные модули

Формальные модули отображают описание требований и связи между ними. DOORS предоставляет удобные возможности навигации по этим связям. Так, можно осуществить переход от зависимого требования к зависящему и наоборот. На рис. 2.9 представлен общий вид формального модуля.

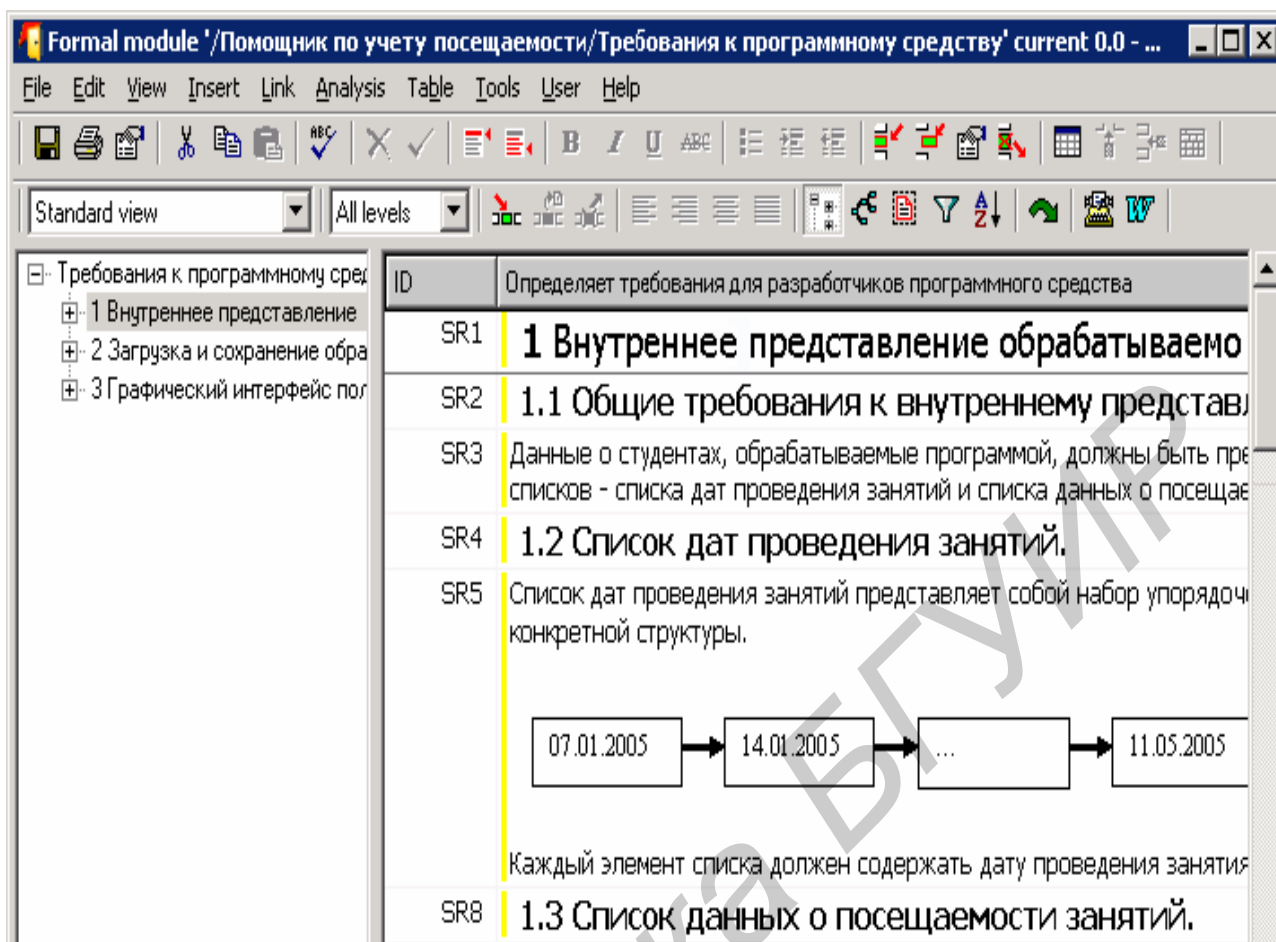


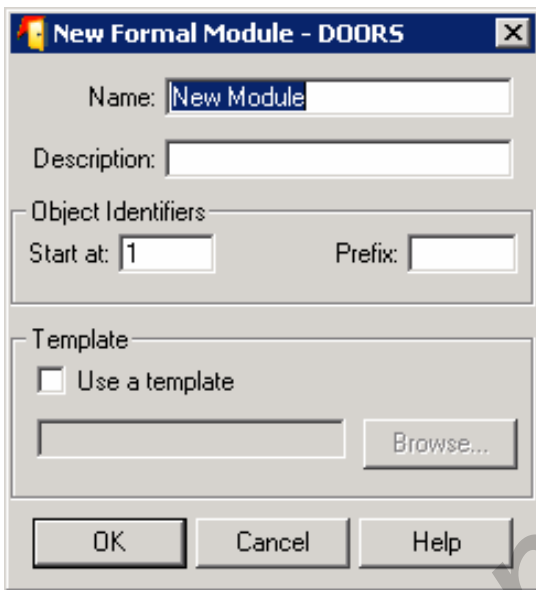
Рис. 2.9. Формальный модуль

Требования, описываемые в формальных модулях, представляются объектами. Такой объект обладает набором свойств и атрибутов. Атрибуты представляют информацию, важную при анализе требований. Их значения отображаются в колонках документа, представляемого формальным модулем.

Представление формального модуля для пользователя содержит проводник на левой панели, отображающий объекты-требования, и детализацию выбранного объекта на правой панели. Все требования пронумерованы. Пример корректной структуры формального модуля представлен на рис. 2.10. Нумерация объектов позволяет уникально определять их положение в иерархии объектов документа. DOORS автоматически поддерживает данную структуру при вырезании, удалении, вставке объекта. Кроме того, все объекты снабжаются уникальными идентификаторами (см. рис. 2.9, колонка ID).

ы документа, р объекта определяет его положение в конкретном документе, ым модулем идентификаторы позволяют уникально описать объекты всех формальных модулей. При удалении объекта его идентификатор остается связанным с ним. Такой подход позволяет эффективно восстановить данный объект или получить информацию о требовании, которое определено удаленным объектом.

Чтобы создать формальный модуль, необходимо в проводнике DOORS выбрать папку (проект), которой будет принадлежать созданный модуль, а затем выбрать соответствующую команду (например выбрать в главном меню File > New > Formal Module). *Опциями*, задаваемыми при создании формального модуля, являются: имя нового модуля; описание нового модуля;



начальный номер идентификатора объекта; префикс идентификатора объекта; настройки шаблона (рис. 2.11). Начальный номер и префикс идентификатора объекта позволяют задать настройки уникальности объектов в создаваемом модуле. Префиксы полезны в тех случаях, когда разрабатываются требования к большим ПС или системам и число формальных модулей велико. Как правило, префикс отражает содержимое модуля. Например, идентификатор UR56 может относиться к формальному модулю требований пользователя (здесь префикс UR – User Requirements).

Рис. 2.11. Создание нового формального модуля

Существует два режима отображения формального модуля – графический и в виде документа. В первом режиме объекты представлены в виде дерева, в котором связь каждого дочернего объекта с объектом–родителем представлена графически. При отображении данных модуля как документа пользователь работает с упорядоченным и структурированным списком объектов. Переключение между режимами может осуществляться посредством специальной кнопки в правом верхнем углу окна формального модуля.

Так как DOORS является многопользовательской системой, она содержит средства синхронизации работы с информацией. Различают *три режима доступа* к формальному модулю: только чтение, совместное редактирование, эксклюзивное редактирование. При открытии модуля можно указать, в каком из режимов будет происходить работа с ним. Также можно изменить режим доступа к уже открытому модулю. Для этого следует выбрать один из режимов в пункте меню Edit > Edit Mode формального модуля.

## 2.3. Работа с требованиями

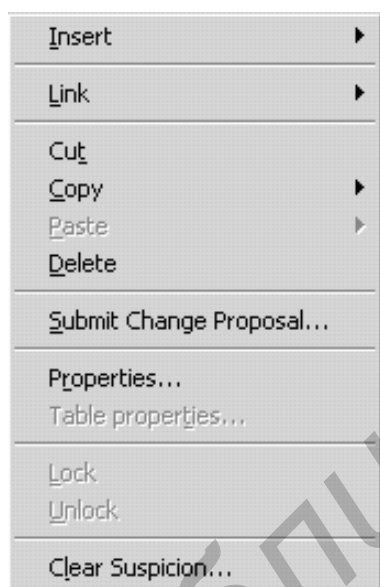
### 2.3.1. Формальные модули как контейнеры объектов

Формальный модуль может рассматриваться как контейнер объектов. Он объединяет объекты в древовидную структуру. Конкретные требования описываются на уровне объектов–листьев дерева. Все объекты, являющиеся объектами–родителями, должны содержать обобщающие описания требований, которые заключены в их дочерних объектах.

С каждым объектом связаны его заголовок, краткий текст описания, основной текст объекта, права доступа, атрибуты и связи с другими объектами.

Заголовок и основной текст отображаются в режиме работы с формальным модулем как с документом. В графическом режиме отображается короткий текст объекта. Не следует создавать объекты, которые содержат и заголовок, и основной текст одновременно. Вместо этого нужно руководствоваться следующим правилом: объекты–родители должны содержать только заголовок, а объекты–листья дерева – только основной текст.

Объект формального модуля можно создать, например, используя контекстное меню и его команду Insert > Below (рис. 2.12).



Так как работа с требованиями тесно связана с обработкой текстовой информации, DOORS предоставляет широкие возможности редактирования текста, а также поддерживает операции работы с OLE объектами, что открывает возможности импорта и экспорта разнородных данных. Редактировать текст объекта можно двумя способами: используя окно свойств объекта и используя «редактирование на месте». Последний способ предполагает изменение объекта непосредственно в окне формального модуля. Для этого необходимо выбрать этот объект двойным нажатием левой кнопки мыши, затем одинарным нажатием указать нужную позицию курсора.

Рис. 2.12. Контекстное меню формального модуля

Так как формальный модуль может содержать большое число объектов, в DOORS реализован механизм, позволяющий установить, какие объекты были изменены, а какие нет.

Так, при отображении каждого объекта наравне с его информацией отображается индикатор его изменений. *Индикатор изменений* – это цветная полоска, расположенная непосредственно возле основной колонки формального модуля, в которой отображается текст объекта. Зеленый цвет полоски означает, что объект не был изменен; желтый – объект был изменен и



эти изменения сохранены; красный – объект изменен, но эти изменения не сохранены; черный – объект был удален, но не очищен.

При изменении объекта и его сохранении DOORS создает для него очередную версию. Это позволяет оперировать объектами разных версий, что является полезным, когда необходимо уточнить, что, когда и кем было изменено в данном объекте.

Помимо поддержки операций над изменяющимися требованиями (функций истории) в DOORS реализованы следующие *функции работы с текстовой информацией*: форматирование (например, представление данных в виде списка, изменение свойств шрифта); вставка специальных символов и URL; проверка орфографии; поиск и замена.

### 2.3.2. Атрибуты

*Атрибут* – это некоторая характеристика объекта, модуля или связи. Например, заголовок объекта и текст объекта – это атрибуты. Атрибуты специфицируют свойства сущности, для которой они определены. Атрибуты позволяют структурировать информацию, относящуюся к данной сущности, а также реализовать концепцию анализа требований.

Различают два типа атрибутов: системные и пользовательские. *Системные атрибуты* – атрибуты, которые создаются DOORS автоматически. *Пользовательские атрибуты* – те, которые определяет пользователь. Системные атрибуты специфицируют системную информацию (например, дату создания модуля, автора последних изменений объекта). В большинстве случаев DOORS следит за установкой их значений, не позволяя пользователям изменять их. Также системные атрибуты отражают обязательную информацию, такую как текст или заголовок объекта. Табл. 2.1 содержит системные атрибуты модуля. Системные атрибуты объектов представляет табл. 2.2.

Таблица 2.1

Системные атрибуты модуля

Имя атрибута	Описание
Created By	Имя пользователя, создавшего модуль
Created On	Дата создания модуля
Description	Дополнительная информация о модуле
Last Modified By	Имя пользователя, который изменял данный модуль последним
Last Modified On	Дата последней модификации модуля
Mapping	Тип схемы связей (например <i>многие ко многим</i> )
Name	Имя модуля
Prefix	Префикс идентификаторов объектов модуля

Каждый атрибут имеет определенный тип значений. *Тип значений* атрибута ограничивает то множество значений, которое может принимать

данный атрибут для конкретных сущностей. В DOORS существуют predefined (базовые) типы (табл. 2.3) и типы, создаваемые пользователем.

Таблица 2.2

Системные атрибуты объектов

Имя атрибута	Описание
Absolute Number	Уникальный идентификатор объекта
Created By	Имя пользователя, создавшего объект
Created On	Дата создания объекта
Created Thru	Тип создания объекта (например копирование)
Last Modified By	Имя пользователя, внесившего последние изменения в объект
Last Modified On	Дата последней модификации объекта
Object Heading	Заголовок объекта
Object Short Text	Короткий текст объекта
Object Text	Текст объекта

Таблица 2.3

Предопределенные типы значений атрибутов

Название типа	Описание
Text	Обычный текст, включая символы новой строки (параграфы)
String	Строковые данные
Integer	Целые числа
Real	Числа с плавающей запятой
Date	Данные о дате
Enumeration	Множество значений атрибута представляется конечным множеством (набором) строковых данных; конкретным значением для сущности является одно из данных значений
Username	Имя пользователя DOORS

Атрибуты применяются при выполнении следующих операций: *фильтрация* – выборка лишь тех объектов, которые отвечают определенному критерию; *сортировка* – сортировка объектов в зависимости от значений атрибутов; *поиск* – в условиях поиска можно задать критерии поиска по атрибутам и их значениям; *статистика атрибутов* – можно просмотреть диаграмму, которая отображает частоту значений атрибута для объектов. Изменять значения атрибутов для объектов можно в окне свойств объекта (рис. 2.13).

Чтобы изменять или просматривать значения атрибутов, можно выбрать в контекстном меню объекта пункт Properties и перейти на закладку Attributes. Для редактирования значений следует выбрать из списка необходимый атрибут и нажать на кнопку Edit. Но этот способ, хотя и является универсальным, не всегда удобен для пользователя. Поэтому DOORS реализует концепцию колонок, когда каждая колонка формального модуля служит для отображения значений атрибутов объектов.

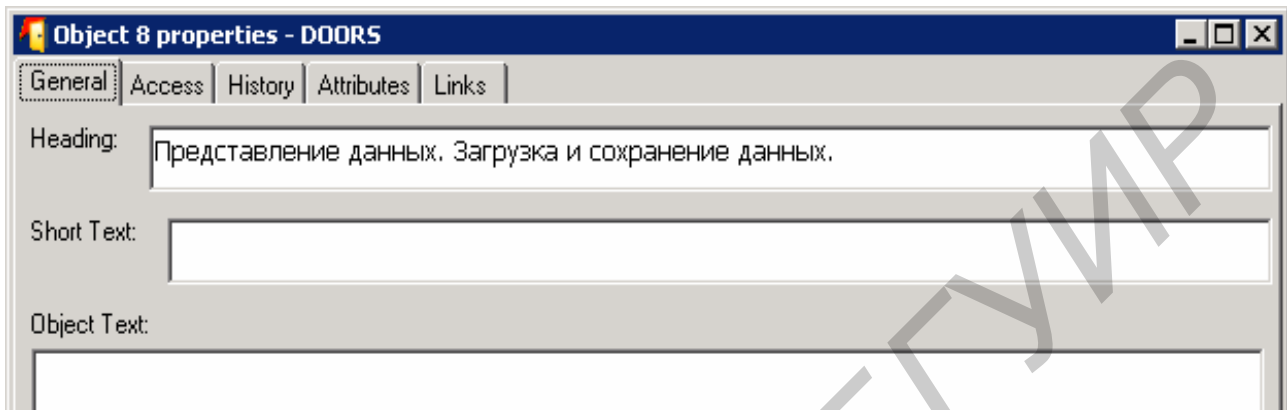


Рис. 2.13. Окно свойств объекта

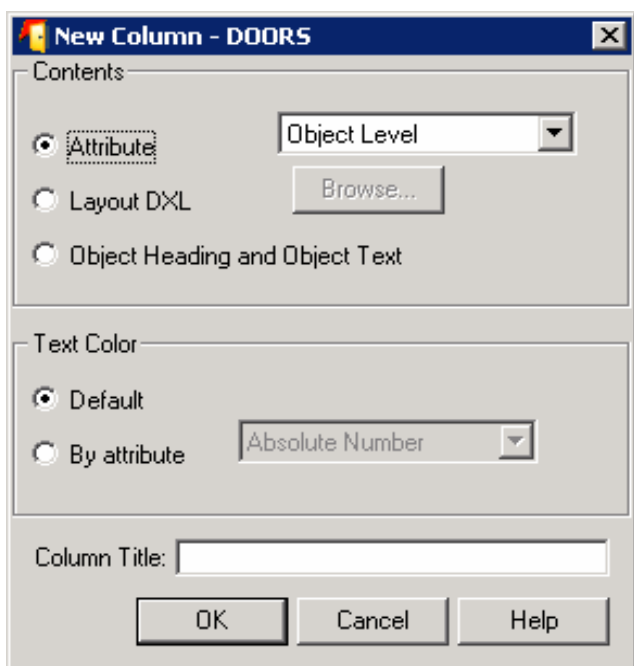
### 2.3.3. Колонки и виды в формальном модуле

DOORS не ограничивает количество атрибутов, которые могут быть определены для сущности. Однако если всегда отображать все имеющиеся атрибуты, то при большом их количестве работать с требованиями будет сложно. Например, если потребуется проводить анализ по атрибуту «Риск», вряд ли есть смысл показывать дату создания объекта. В силу этого для атрибутов объектов DOORS реализует концепцию колонок. **Колонка** служит для отображения значений атрибута объектов того модуля, в котором она определена. Важно понимать, что физически информация в базе данных хранится в виде атрибутов; и независимо от того, видима колонка или нет, создана колонка или удалена, значение атрибута продолжает свое существование. Колонки обеспечивают удобную для пользователя работу с объектами, поскольку позволяют устанавливать значение атрибута для объекта непосредственно из окна формального модуля.

В зависимости от отображаемой информации, различают *три вида колонок*: колонки, отображающие значения атрибутов для конкретных объектов; колонки, отображающие данные, которые получены с использованием DXL; колонки, отображающие заголовок и текст объекта.

Чтобы создать новую колонку для формального модуля, можно выбрать опцию вставки колонки в главном меню окна формального модуля (Insert > Column). Каждая колонка имеет свое заглавие (рис. 2.14). Для колонок, отображающих значения атрибутов, заглавием по умолчанию является название атрибута. Для колонок, отображающих текст и заголовок объектов, по умолчанию отображается текст описания формального модуля.

Чтобы работать со всеми атрибутами, число которых в реальных системах может превышать 32, в DOORS поддерживаются виды. *Вид* – это то представление формального модуля, с которым работает пользователь.



Вид включает набор отображенных колонок, наложенные фильтры, сортировки, текущий объект, позицию и размеры окна формального модуля. Виды удобны тем, что они могут быть связаны с конкретными пользователями. Так, можно установить опции, согласно которым некоторый вид станет видом по умолчанию формального модуля для какого-либо пользователя либо для всех пользователей.

Рис. 2.14. Создание колонки для отображения значений атрибута

### 2.3.4. Поиск, фильтрация и сортировка

DOORS имеет мощные средства, позволяющие осуществлять поиск и замену данных, переход к нужным объектам, настраивать фильтры и сортировки.

DOORS позволяет находить информацию двух видов: проекты, папки, модули; текст в модуле. *Поиск проектов, папок и модулей* осуществляется из окна проводника DOORS. Для активизации функций поиска можно выбрать пункт меню Tools > Find. Окно поиска представлено на рис. 2.15. Табл. 2.4 содержит критерии поиска проектов, папок и модулей.

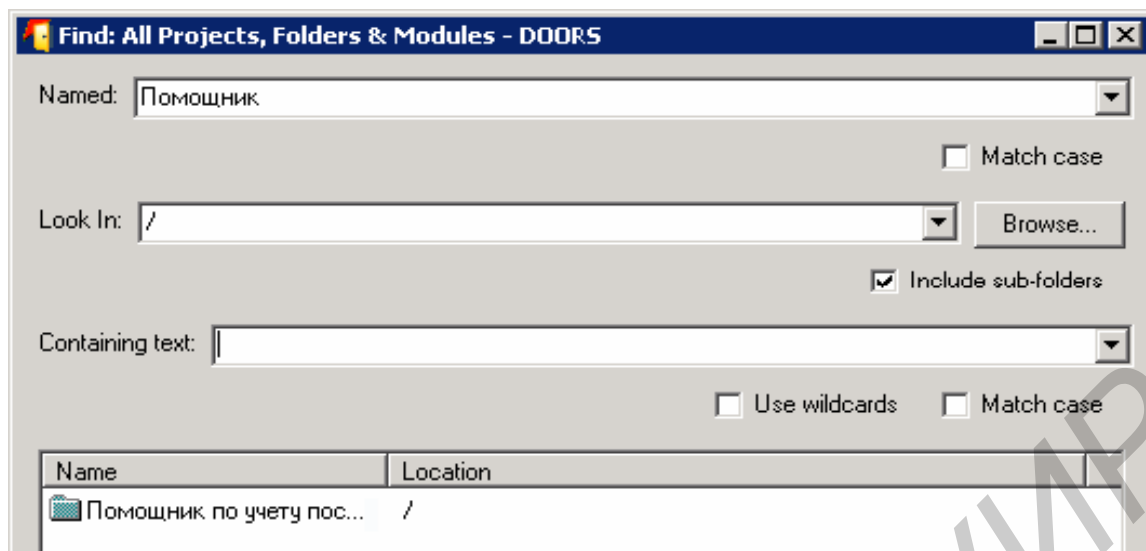


Рис. 2.15. Поиск проектов, папок и модулей

Таблица 2.4

Критерии поиска проектов, папок и модулей

Критерий	Описание
Имя проекта, папки, модуля	Имя проекта, папки или модуля, которые необходимо найти. Имя может содержать как полное имя, так и его фрагмент. Символ «?» соответствует одному произвольному символу, «*» – последовательности символов произвольной длины
Папка или проект, в котором необходимо осуществлять поиск	Путь к папке или проекту, в котором необходимо осуществлять поиск. Можно задать опцию поиска во вложенных папках и проектах
Текст, который должен содержаться в данных, находящихся в проекте, папке, модуле	Тот текст, который должен содержаться в данных, находящихся в проекте, папке, модуле
Текстовые опции поиска	Допустимо задание двух опций: использование групповых символов и учет либо игнорирование регистра символов при поиске

*Поиск текста в модуле* позволяет найти конкретные объекты, существующие в данном модуле. Для активизации этой функции можно в

главном меню окна модуля выбрать пункт Edit > Find. Такой поиск может быть осуществлен по значениям атрибутов модуля.

В качестве шаблона поиска можно указать регулярное выражение. Для более точного формулирования критериев можно воспользоваться возможностями расширенного поиска. На рис. 2.16 изображено окно поиска текста.

Функции замены активизируются посредством выбора команды Edit > Replace в главном меню модуля. Опции замены позволяют заменять указанный для поиска фрагмент как пошагово, так и для всех найденных фрагментов (рис. 2.17).

Команда главного меню модуля Edit > Go To выполняет переход к указанному объекту. Объект может быть указан как при помощи абсолютного номера, так и при помощи номера секции. Например, значение 58 будет воспринято как порядковый номер объекта, а значение 5.8 – как номер секции 5, содержащей объект 8, к которому надо перейти. На рис. 2.18 изображено окно, из которого осуществляется переход к объекту формального модуля.

Фильтры позволяют отобразить лишь те объекты, которые отвечают определенному набору критериев поиска. Фильтры удобны при проведении анализа требований. Они могут быть настроены и применены с помощью команды главного меню модуля Tools > Filter. Различают два вида фильтров: простые и расширенные.

В табл. 2.5 сведены основные возможности фильтров. Общий вид окна настройки фильтра представлен на рис. 2.19.

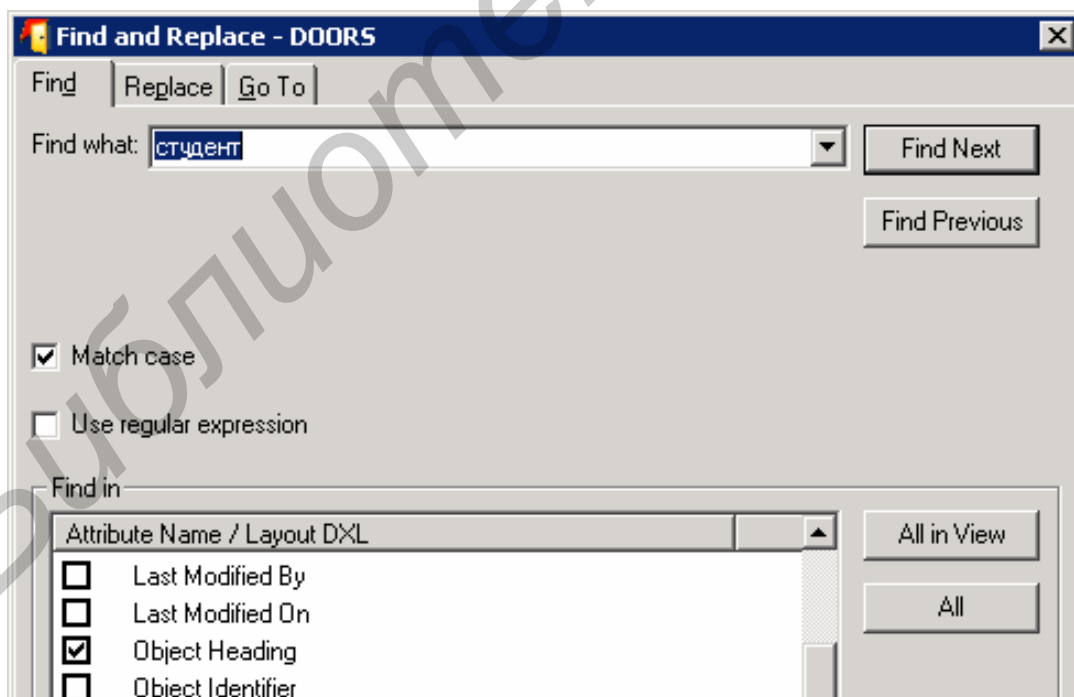


Рис. 2.16. Поиск текста



Рис. 2.17. Замена текста

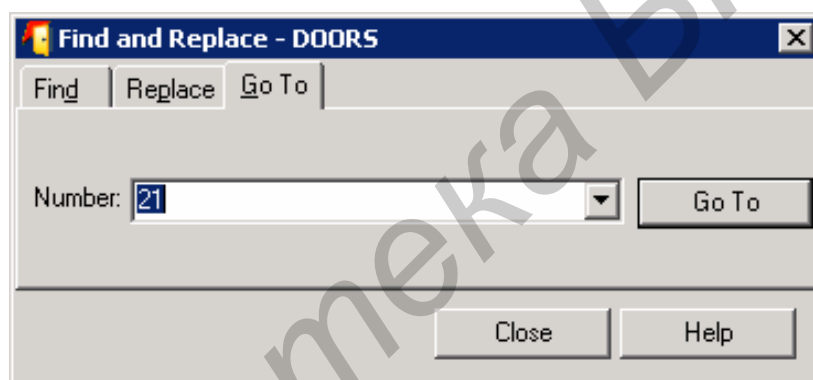


Рис. 2.18. Переход к объектам в формальном модуле

Таблица 2.5

Возможности простых и расширенных фильтров

Простые фильтры	Расширенные фильтры
<p>фильтруют:</p> <ul style="list-style-type: none"> <li>– по содержанию любого атрибута типа текст или строка;</li> <li>– по номеру объекта;</li> <li>– по содержанию любой колонки;</li> <li>– по значению одного атрибута любого типа;</li> <li>– по признаку наличия связей у объекта;</li> <li>– по признаку, является ли объект</li> </ul>	<ul style="list-style-type: none"> <li>– комбинируют простые фильтры путем описания правил на специальном языке;</li> <li>– задают опции фильтра, управляющие выводом отображения (например, можно задать, будут ли выводиться на экран только те объекты, которые удовлетворяют условию фильтра, или же будут отображаться также</li> </ul>

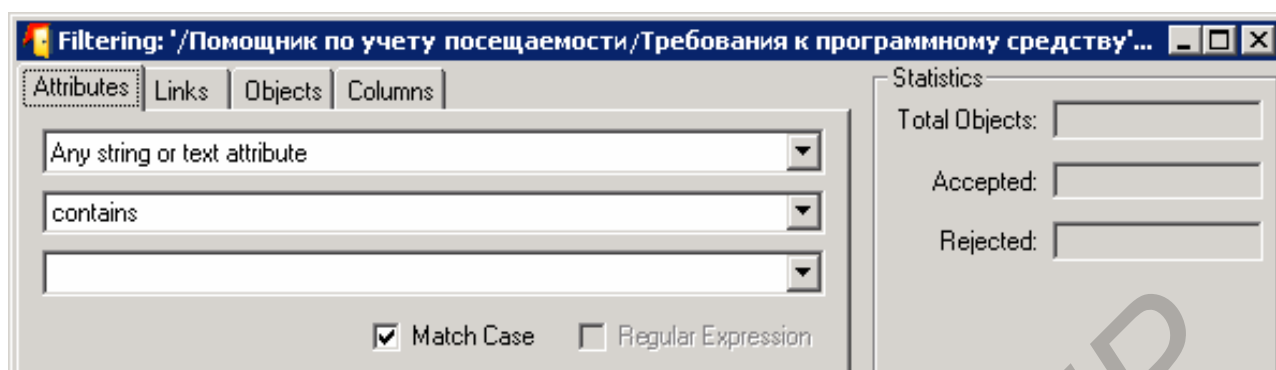


Рис. 2.19. Окно настройки фильтра

Сортировки изменяют порядок отображения объектов в модуле. Этот порядок определяется значениями атрибутов объектов. Сортировка является *простой*, если все объекты упорядочиваются по значениям какого-либо одного атрибута, и *расширенной*, если на порядок объектов влияют значения нескольких атрибутов. Для задания сортировки используют команду главного меню модуля Tools > Sort. Окно задания критериев сортировки представлено на рис. 2.20.

*Простая сортировка* имеет два критерия: атрибут, по значениям которого упорядочиваются объекты; порядок упорядочивания (восходящий либо нисходящий). *Расширенная сортировка* представляет собой набор простых сортировок. Например, при применении расширенной сортировки по атрибутам «Приоритет» и «Риск» все объекты сортируются по значениям атрибута «Приоритет». Объекты, имеющие одинаковые значения данного атрибута, в свою очередь еще раз сортируются, исходя из значений атрибута «Риск». Таким образом, образуются отсортированные группы.

### 2.3.5. Связи между требованиями

Во всех реальных проектах требования взаимосвязаны. Зависимости могут существовать как между требованиями разных уровней (например между системными требованиями и требованиями пользователя), так и между требованиями одного уровня (например в рамках системных требований). Управление требованиями основано именно на механизме связей.



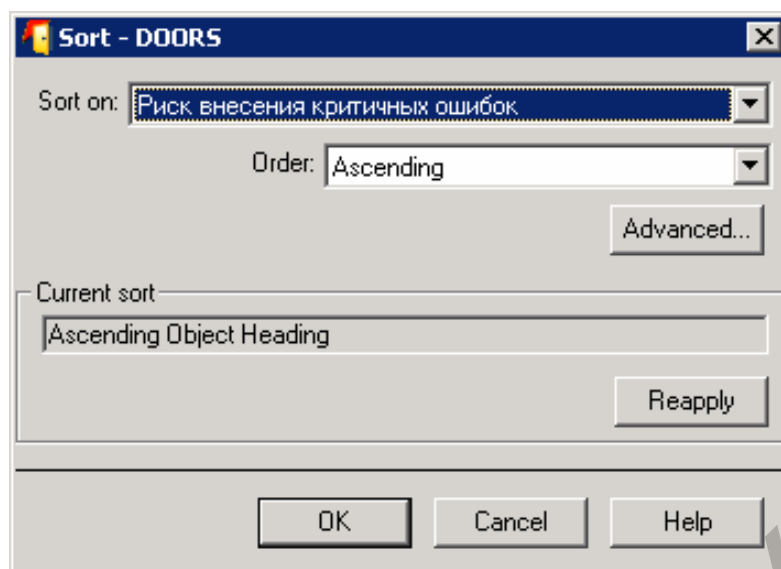


Рис. 2.20. Окно задания критериев сортировки

Так как требования в DOORS определяются объектами, то связи могут быть применены лишь к данным объектам. Связи в DOORS являются однонаправленными, то есть для любых двух связанных объектов один всегда является объектом-источником (начальным объектом), от которого зависит второй объект, а второй – целевым (конечным) объектом. Для объекта-источника связь является исходящей, для целевого объекта – входящей (рис. 2.21).

Существует несколько *способов создания связей* из формального модуля: посредством перетаскивания объектов (используя принцип «drag and drop»); посредством точного задания начала и конца связи; посредством расширенных функций создания связей. Первый способ позволяет определять один объект-источник и один целевой объект. Чтобы создать связь этим способом, необходимо выделить начальный объект и, не отпуская кнопки мыши, переместить курсор на тот объект, который должен стать конечным. После отпускания кнопки мыши появится диалоговое окно, предлагающее создать связь.

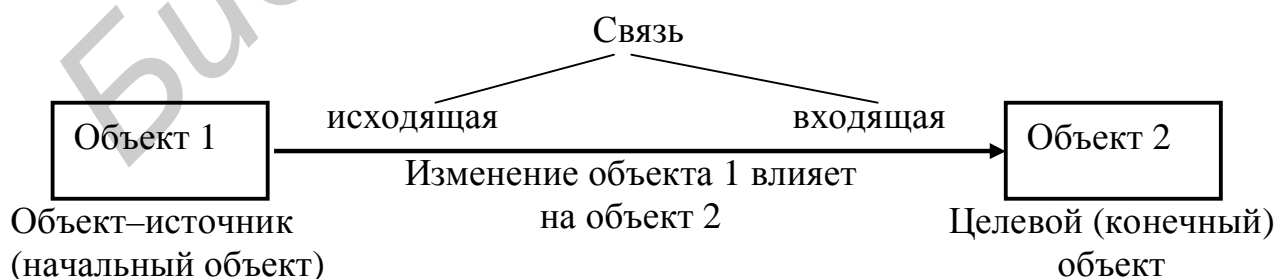


Рис. 2.21. Терминология связей DOORS

Создание связей посредством точного задания их начала и конца является универсальным способом. Этот способ позволяет определять один объект–источник и несколько целевых объектов. Для такого связывания необходимо выделить нужный объект и в главном меню окна формального модуля выполнить команду Link > Start Link. Далее следует выделить те объекты, которые должны стать конечными, и выполнить команду меню Link > Make Link From Start (рис. 2.22).

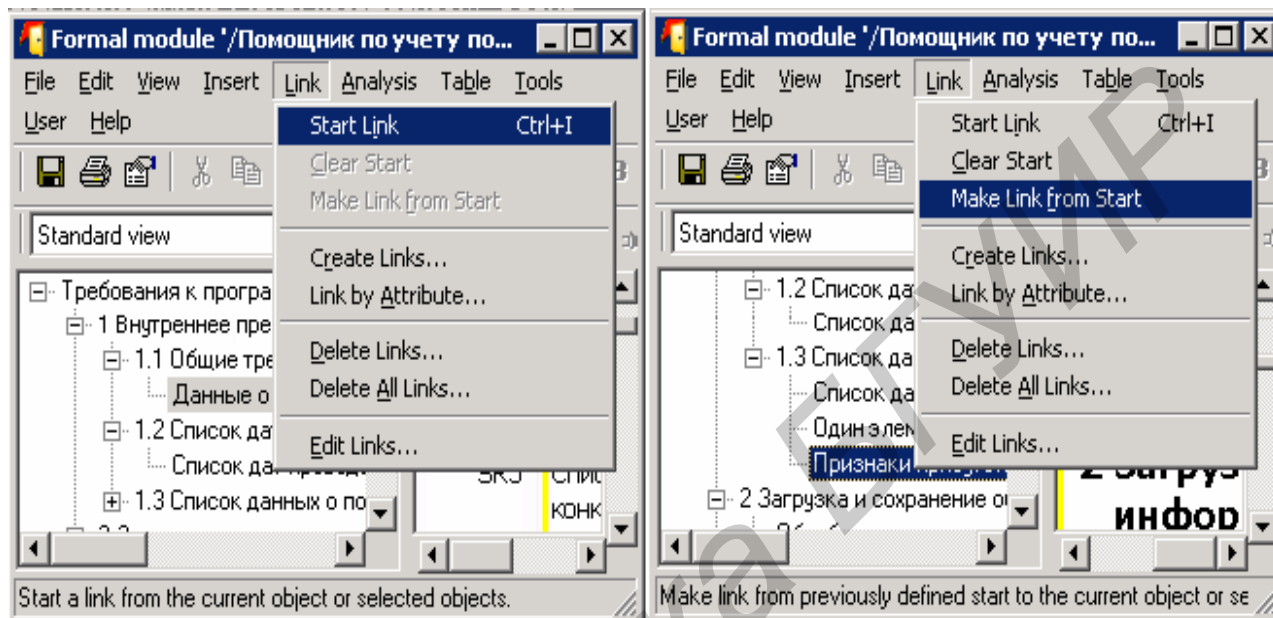


Рис. 2.22. Создание связей точным заданием начала и конца связи

Объекты–источники продолжают быть выделенными и после создания связи. Это удобно, если проект имеет развитую структуру связей, так как пользователю не приходится выделять одни и те же объекты постоянно для создания очередной связи. Чтобы очистить объекты–источники, следует выполнить команду меню Link > Clear Start. Данная команда снимает пометку о том, что при создании новых связей выделенные объекты будут являться объектами–источниками.

Расширенные функции создания связей могут быть активированы путем выполнения команд главного меню формального модуля Link > Create Links и Link > Link by Attribute. Как и объекты, связи могут содержать атрибуты. Все связи объекта доступны на закладке Links его свойств (рис. 2.23).

In/Out	Module	Baseline	Object Heading/Text	ID	Link Module	Link Module Baseli...
Out	/Помощник по учету п...	Current	Должна обеспечивать...	15	/Помощн...	Current
Out	/Помощник по учету п...	Current	При изменении пользо...	33	/Помощн...	Current
Out	/Помощник по учету п...	Current	Инициализация таблиц...	32	/Помощн...	Current

Рис. 2.23. Связи между объектами в окне свойств объекта

Для изменения или удаления связи можно выполнить команду главного меню формального модуля Link > Edit Links.

### 2.3.6. Подозрительные связи

При изменении одного объекта автоматически изменяются его связи с другими объектами. По отношению к этим объектам данные связи становятся *подозрительными*. Подозрительные связи отмечаются особым образом (посредством отображения вопросительного знака в окне формального модуля возле объекта, по отношению к которому связь является подозрительной). Независимо от того, изменяется начальный или конечный объект, связь становится подозрительной лишь для неизменившегося объекта (рис. 2.24).

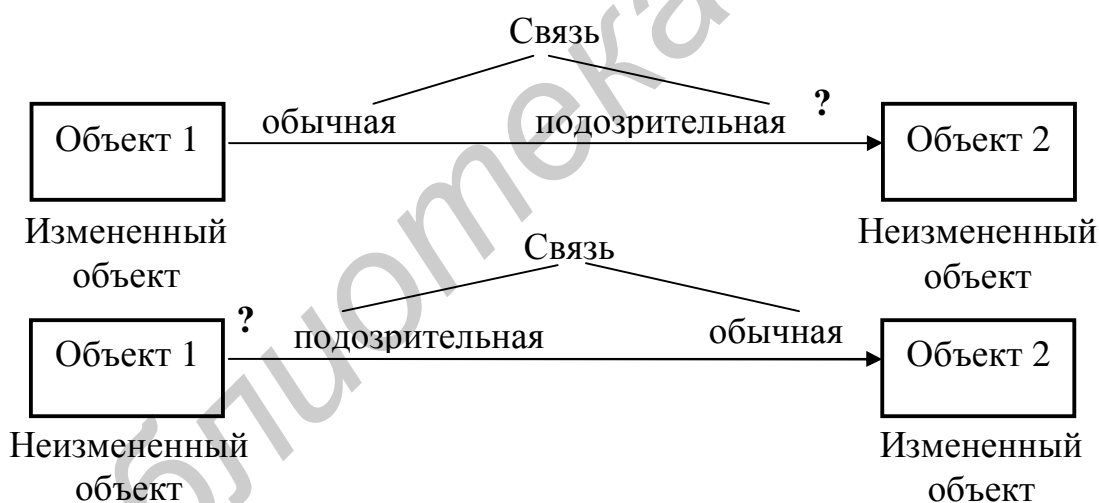


Рис. 2.24. Подозрительные связи

Все опции работы с подозрительными связями доступны из главного меню окна формального модуля (Analysis > Suspect Links).

Существует два способа снятия пометки о подозрительности связи:

- при помощи инициализации подозрительных связей; реализуется командой главного меню Analysis > Suspect Links > Initialize; позволяет отменить подозрение для всех связей данного формального модуля;

- при помощи очистки подозрительных связей; реализуется командами Analysis > Suspect Links > Clear или Analysis > Suspect Links > Clear All; позволяет снимать подозрения с отдельных связей (рис. 2.25).

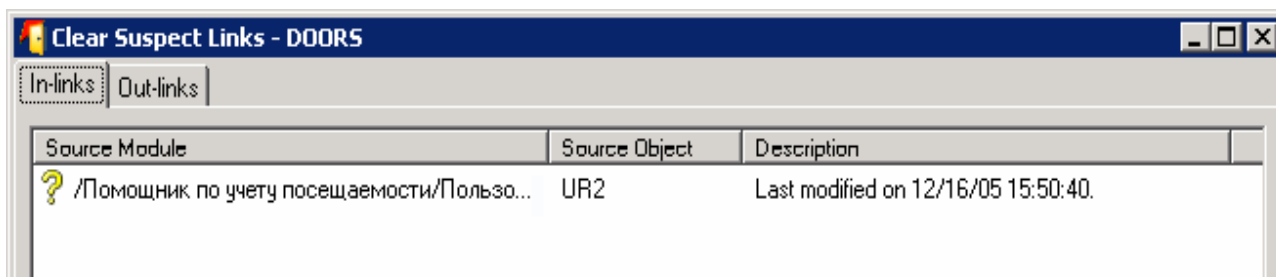


Рис. 2.25. Снятие пометки о подозрительности связи

### 2.3.7. Модули связей

Формальные модули хранят информацию о требованиях и связях между ними. Однако сами связи в DOORS хранятся в отдельных модулях связей. Каждый *модуль связей* содержит связи определенного типа, устанавливаемого пользователем (пользователь определяет признаки, по которым он классифицирует связи). Например, можно определить модуль связей «Связи трассируемости» и отнести в него те связи, которые отражают зависимости между результатами тестирования, системными требованиями и требованиями пользователя.

По умолчанию все связи сохраняются в модуле связей «DOORS Links», создаваемом автоматически в текущей папке или проекте. Обычно этого модуля связей достаточно. Создавать свои модули связей имеет смысл лишь в тех случаях, когда количество связей велико и их структура достаточно сложна.

Для структуризации информации в модулях связей хранятся не сами связи, а их наборы. *Набор связей* содержит информацию о связях одного формального модуля с другим. Все связи в наборе имеют одно направление: все объекты–источники набора связей принадлежат одному модулю, а целевые объек-

ты – другому. На рис. 2.26 отображен пример представления модуля связей в DOORS. Черные клеточки обозначают наличие связей между объектами.

DOORS позволяет управлять типом и направлением связей, создаваемых всеми пользователями. Посредством управления наборами связей можно задавать разрешенные связи между модулями и запрещать любые другие комбинации. DOORS позволяет задавать модули связей по умолчанию для той или иной пары формальных модулей, структурируя таким образом сохраняемую информацию о связях.

Чтобы создать модуль связей, можно выполнить команду File > New > Link Module главного меню окна DOORS. При создании модуля необходимо определить схему связи между объектами, которая будет поддерживаться данным модулем. Существуют следующие *схемы связей объектов*: *многие ко многим* (каждый объект может иметь несколько исходящих и несколько входящих связей); *многие к одному* (каждый объект может иметь несколько исходящих связей и одну входящую); *один ко многим* (каждый объект может

иметь одну исходящую связь и несколько входящих); *один к одному* (каждый объект может иметь только одну исходящую связь и только одну входящую).



Рис. 2.26. Модуль связей

Для создания набора связей можно в главном меню модуля связей выполнить команду **File > New > Linkset**. Если требуется дополнительный контроль над связями, можно задать дополнительные опции. Так, если создать набор связей посредством его добавления из окна свойств формального модуля (**File > Module Properties > Linksets > Add**), то можно установить использование данного модуля связей обязательным, принудив тем самым всех пользователей сохранять информацию о связях в данном модуле независимо от того, какие модули связей заданы для них по умолчанию. Для определения такого поведения необходимо установить флаг **Mandatory** при создании набора связей. Если установить флаг **Overrideable**, все связи создаваемого набора связей будут сохраняться в модуле связей по умолчанию активного пользователя (рис. 2.27).

DOORS позволяет установить запрет на использование исходящих связей и ограничить использование входящих. Это бывает удобно, если существует большое количество формальных модулей со сложными взаимосвязями или если этого требует управление проектом. Например, часто запрещают использовать исходящие связи для формальных модулей, описывающих требования пользователя. Опции, определяющие такие настройки, устанавливаются из окна свойств формального модуля (**File > Properties**).

### 2.3.8. Средства анализа связей

Анализ связей позволяет определить влияние, которое может оказать изменение тех или иных требований. Все *инструменты анализа связей* доступны из меню Analysis формального модуля. Этими инструментами являются: инструмент анализа влияния; инструмент анализа трассируемости; мастер анализа; проводник трассируемости; инструмент фильтрации по связям; средства управления подозрительными связями.

*Анализ влияния* позволяет оценить влияние, которое оказывают требования в данном формальном модуле на другие требования. Анализ влияния предоставляет информацию об исходящих связях анализируемого модуля.

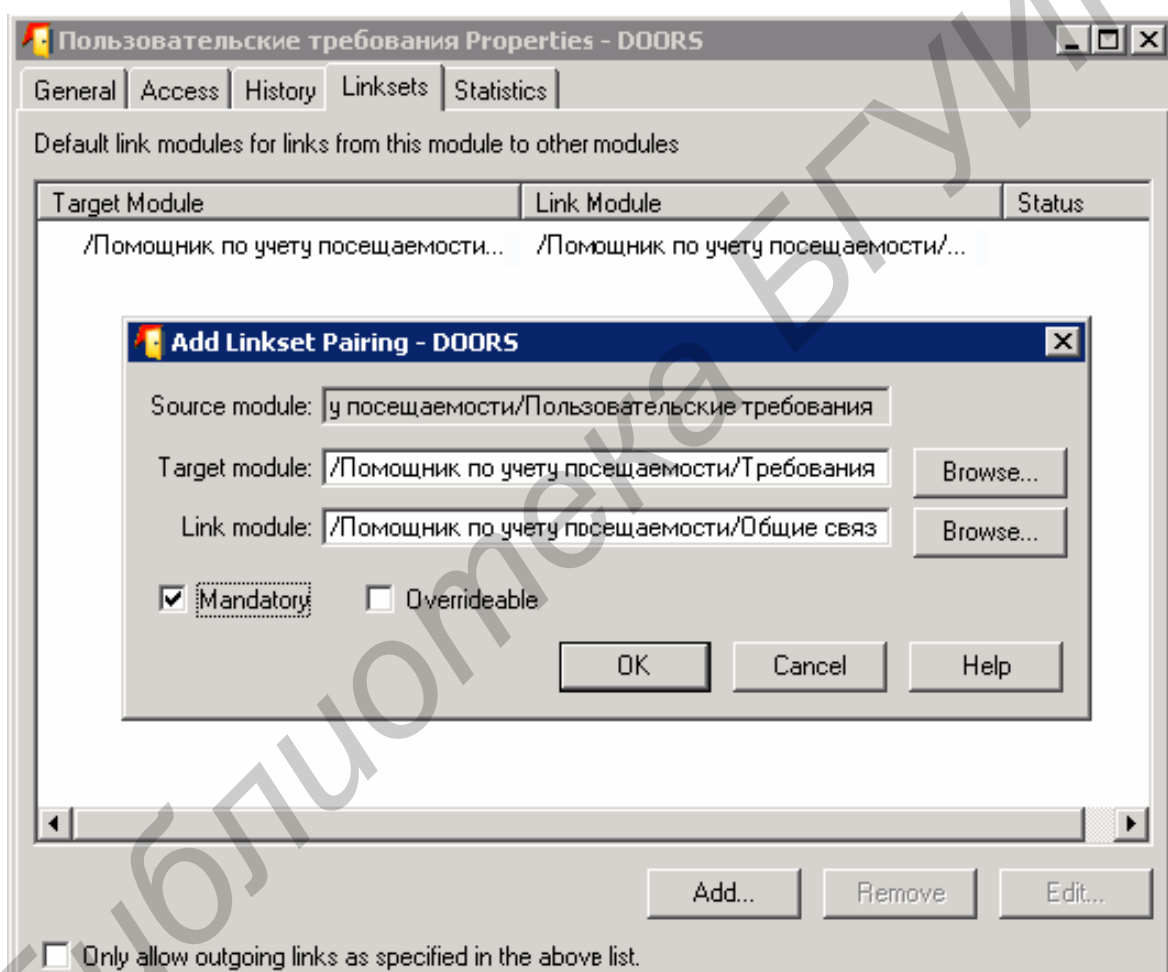


Рис. 2.27. Ограничение использования связей для формальных модулей

*Анализ трассируемости* (анализ последствий) дает возможность определить влияние изменения требований, описанных в других формальных модулях, на требования данного формального модуля. Анализ трассируемости предоставляет информацию о входящих связях анализируемого модуля.

Для запуска анализа влияния необходимо выполнить команду Analysis > Impact главного меню формального модуля. В настройках можно задать глубину проведения анализа (рис. 2.28). По умолчанию ее значение равно 1.

Это соответствует участию в анализе лишь требований, непосредственно связанных с требованиями в данном формальном модуле. Если установить значение глубины 2, то в анализе будут принимать участие как непосредственно связанные объекты (объекты уровня 1), так и объекты других формальных модулей, которые связаны с объектами уровня 1.

Для запуска анализа трассируемости выполняется команда Analysis > Trace главного меню формального модуля. При этом также необходимо задать глубину анализа в окне, аналогичном представленному на рис. 2.28.

Результатом проведения анализа влияния и анализа трассируемости является дополнительная колонка, которая добавляется к виду формального модуля. Информация, содержащаяся в ячейках этой колонки, относится к объекту другого формального модуля, который связан с текущим объектом.

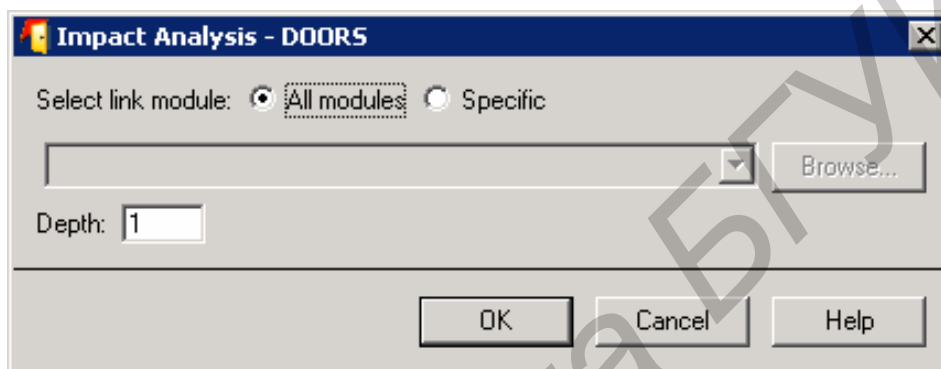


Рис. 2.28. Настройка анализа влияния

*Мастер анализа* позволяет поэтапно задать детальную информацию о проводимом анализе (табл. 2.6). Мастер может быть запущен посредством выполнения команды меню Analysis > Wizard.

Результатом анализа, проведенного с помощью мастера, является набор колонок, которые были определены при настройке анализа. Главное отличие этого результата от результатов, получаемых при использовании средств анализа влияния и анализа трассируемости, состоит в возможностях более детальной настройки отображаемой информации и более точного задания данных, которые будут участвовать в анализе.

*Фильтрация по связям* позволяет отфильтровать (вывести) лишь те объекты, данные о связях которых подчиняются некоторым правилам. Фильтрация по связям – это одна из опций фильтров. Сам фильтр может быть настроен командой главного меню Tools > Filter формального модуля. На закладке Links находятся опции фильтрации по связям (рис. 2.30).

*Проводник трассируемости* (рис. 2.31) предоставляет удобный интерфейс для навигации по имеющим связи объектам. Он может быть запущен из главного меню формального модуля (Analysis > Traceability Explorer). Меню View проводника содержит опции настройки отображаемой информации.

## Этапы настройки анализа, проводимого с помощью мастера анализа

Номер этапа	Название этапа	Описание этапа
1	Границы анализа	Задание формальных модулей и модулей связей, которые участвуют в анализе
2	Тип анализа	Задание связей, которые участвуют в анализе (анализировать можно либо входящие, либо исходящие связи)
3	Атрибуты	Задание атрибутов, которые будут отображены в колонках формального модуля (рис. 2.29)
4	Другая информация	Задание информации, касающейся отображаемой информации, а также установка опции рекурсивного анализа
5	Рекурсия	Задание колонок, которые должны быть отображены при рекурсивном анализе; эти колонки будут содержать информацию о связанных объектах

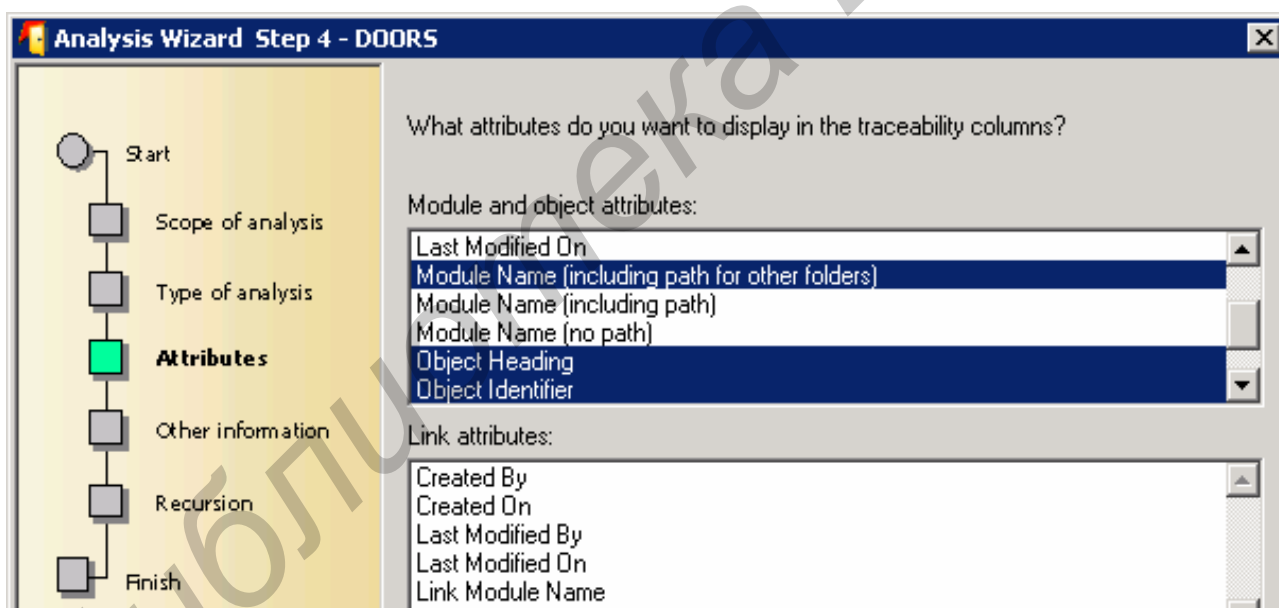


Рис. 2.29. Настройка мастера анализа: этап задания атрибутов

Средства управления подозрительными связями позволяют отображать информацию о подозрительных связях, снимать подозрения со связей, отфильтровывать лишь объекты с подозрительными связями в формальном модуле. Команда главного меню Analysis > Suspect Links > Display All Changes формального модуля добавляет колонку к виду формального модуля, содержащую информацию обо всех изменениях связанного объекта. Команда Analysis > Suspect Links > Display Last Changes добавляет колонку, которая



содержит информацию лишь о последних изменениях связанного объекта.

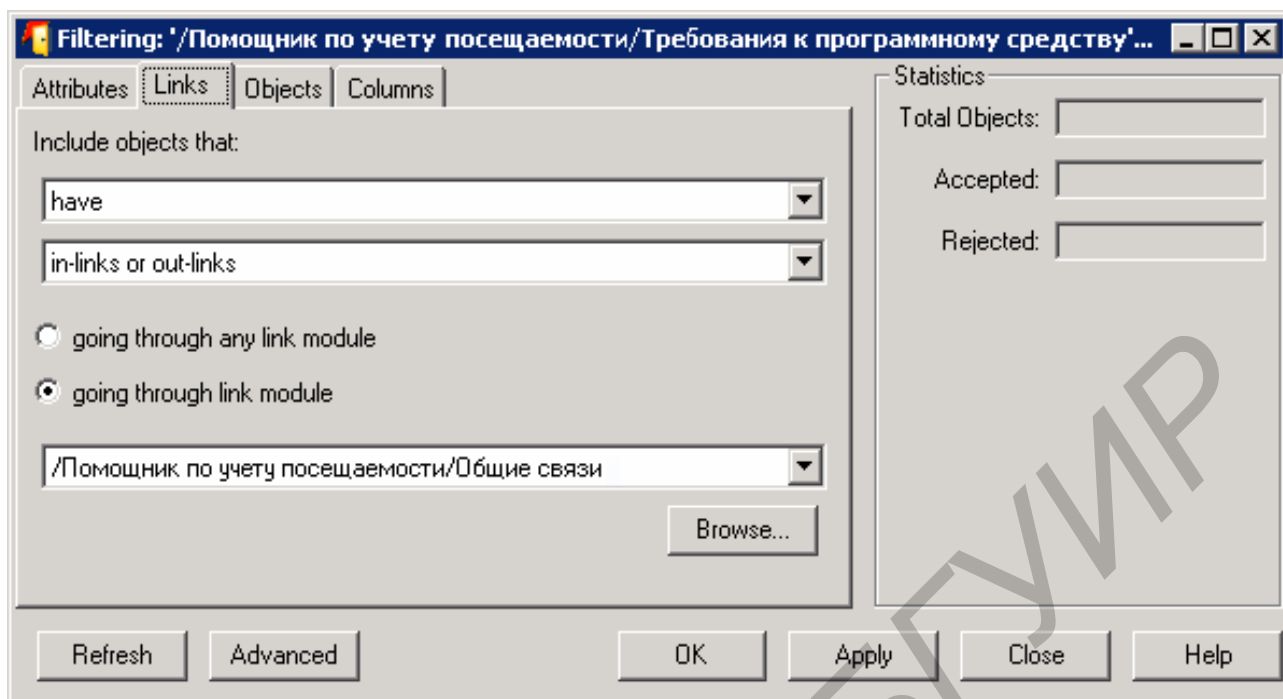


Рис. 2.30. Фильтрация по связям

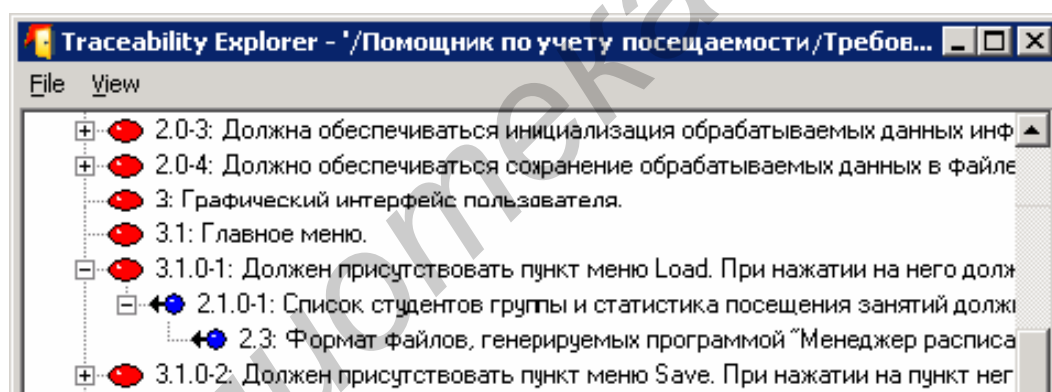


Рис. 2.31. Проводник трассируемости

Командой Analysis > Suspect Links > Display Indicators можно установить видимость индикаторов подозрительных связей. Команды Clear, Clear All, Initialize позволяют снять пометки о подозрительности связей. Команды меню управления подозрительными связями представлены на рис. 2.32.

Фильтр подозрительных связей запускается посредством выполнения команды главного меню Analysis > Suspect Links > Filter. Результатом его применения является набор объектов, имеющих подозрительные связи.

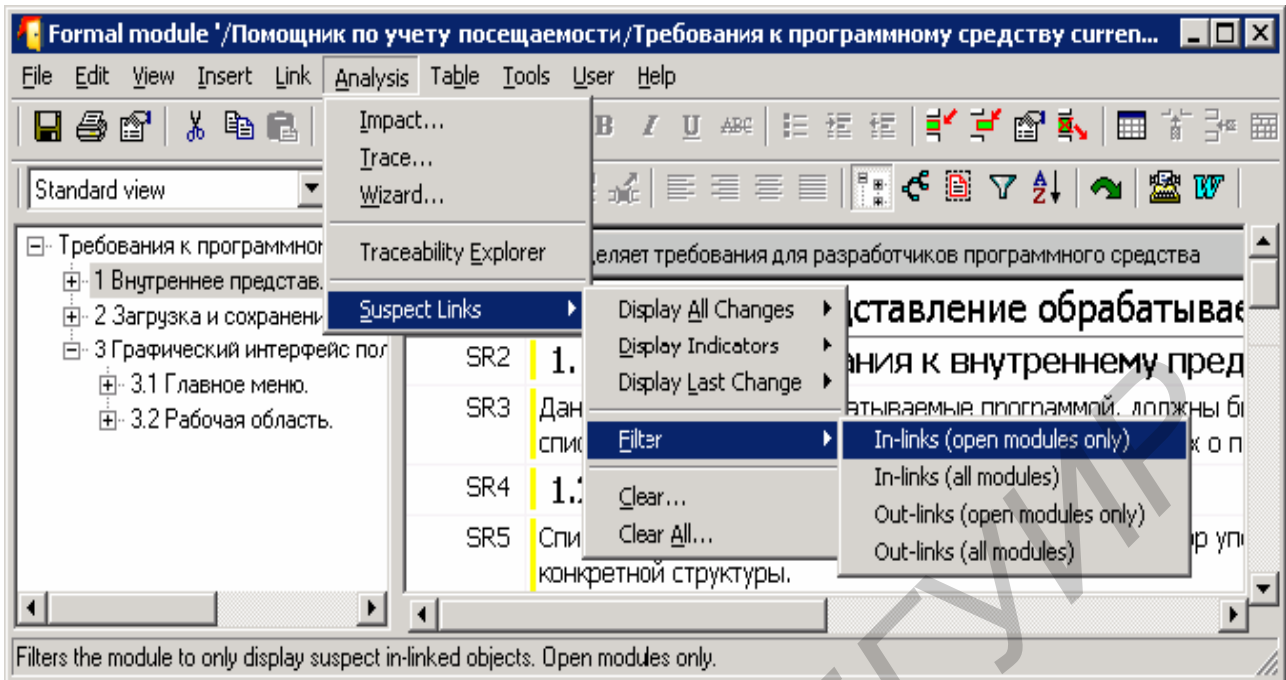


Рис. 2.32. Управление подозрительными связями

## 2.4. Управление изменениями требований

### 2.4.1. Инструменты управления изменениями требований

Набор механизмов управления изменениями требований носит административный характер. Он позволяет координировать процесс изменения требований. Набор включает следующие *инструменты*: базовые версии; цифровая подпись; система предложений о внесении изменений.

### 2.4.2. Базовые версии

*Базовая версия* формального модуля – это его версия, которая доступна пользователям только для чтения. Каждая базовая версия определяет завершение очередного этапа разработки требований. Не следует создавать базовые версии, если высока вероятность дальнейшей модификации требований данного модуля. Базовая версия формального модуля создается не по истечении определенного количества времени, а по мере утверждения текущих требований или накануне внесения значительных изменений.

При создании новой базовой версии происходит сохранение истории данного формального модуля. *История* включает информацию обо всех определениях и типах атрибутов, а также об объектах, которые были созданы, удалены или изменены, начиная с последней базовой версии модуля. После создания базовой версии история формального модуля начинает создаваться заново. Все изменения модуля доступны в его базовых версиях.

Чтобы просмотреть базовые версии, необходимо выполнить команду главного меню формального модуля File > Baseline > View. В появившемся списке следует выбрать необходимую базовую версию и нажать кнопку Open.

Над базовыми версиями определены *операции* создания, удаления, чтения (просмотра), копирования, сравнения. Возможно создание базовых версий для группы модулей. Операции над базовыми версиями доступны из меню File > Baseline. На рис. 2.33 представлен вид этого меню.

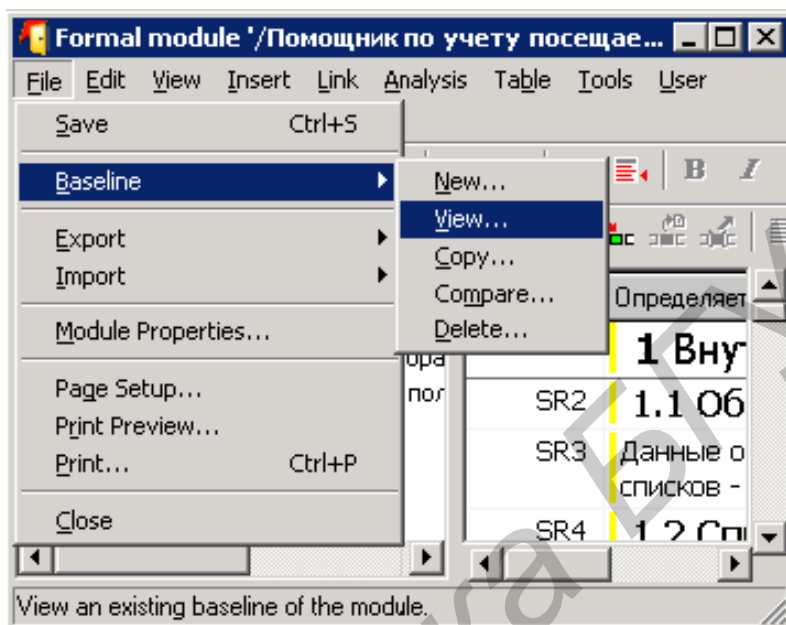


Рис. 2.33. Меню работы с базовыми версиями

### 2.4.3. Цифровая подпись

DOORS – это многопользовательская система. Чтобы вся работа велась согласованно, информация о состоянии проекта находилась в одном месте и все пользователи могли увидеть ее, в DOORS поддерживаются цифровые подписи. *Цифровая подпись* удостоверяет, что документ подписан определенным лицом; не дает возможности лицу, поставившему эту подпись, отказаться от нее; гарантирует целостность подписанного текста.

Так как базовая версия формального модуля – это логически заверченный набор требований, то подписываются именно базовые версии. Текстом цифровой подписи являются краткая метка (например, «Содержание модуля одобрено», «Отклонено») и описание, которое может содержать произвольный текст. Чтобы просмотреть или поставить цифровую подпись, необходимо выполнить команду главного меню File > Baseline > View формального модуля. Далее следует выбрать интересующую базовую версию и нажать кнопку Signatures. Окна, с которыми работает пользователь при постановке или просмотре цифровой подписи, представлены на рис. 2.34.

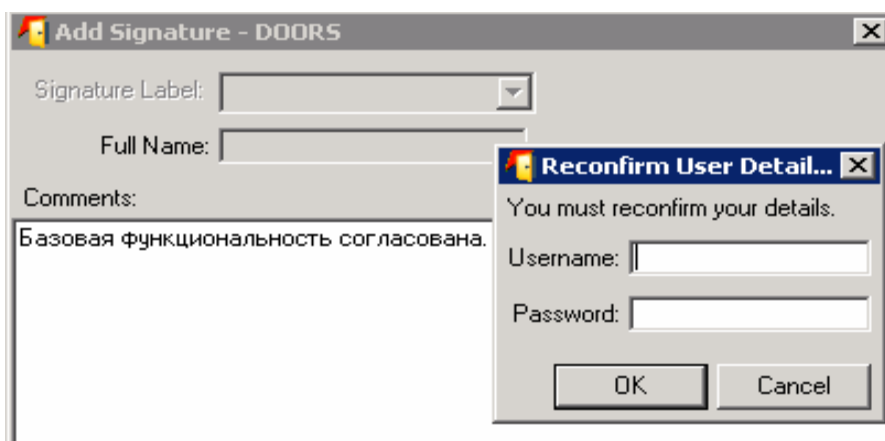


Рис. 2.34. Работа с цифровой подписью

#### 2.4.4. Система предложений о внесении изменений

Для решения вопросов, связанных с внесением, доработкой, отклонением требований, DOORS предоставляет систему предложений о внесении изменений. Система предложений о внесении изменений позволяет одним пользователям лишь вносить предложения об изменении требований, а другим – принимать решения об их внесении. В DOORS различают два типа предложений:

- предложение о внесении изменения – это подробное предложение об изменении конкретного объекта в конкретном модуле (например предложение об изменении значения атрибута);
- рекомендация – это предложение об изменении более высокого уровня (например предложение о добавлении некоторой функции ПС).

При создании предложения о внесении изменения необходимо указать причину изменения, тип изменения (например модификация) и приоритет, с которым должно рассматриваться данное предложение. Вся эта информация используется при анализе предложений о внесении изменений. Все операции работы с системой предложений о внесении изменений доступны из главного меню окна DOORS (Tools > Change Proposal System, рис. 2.35).

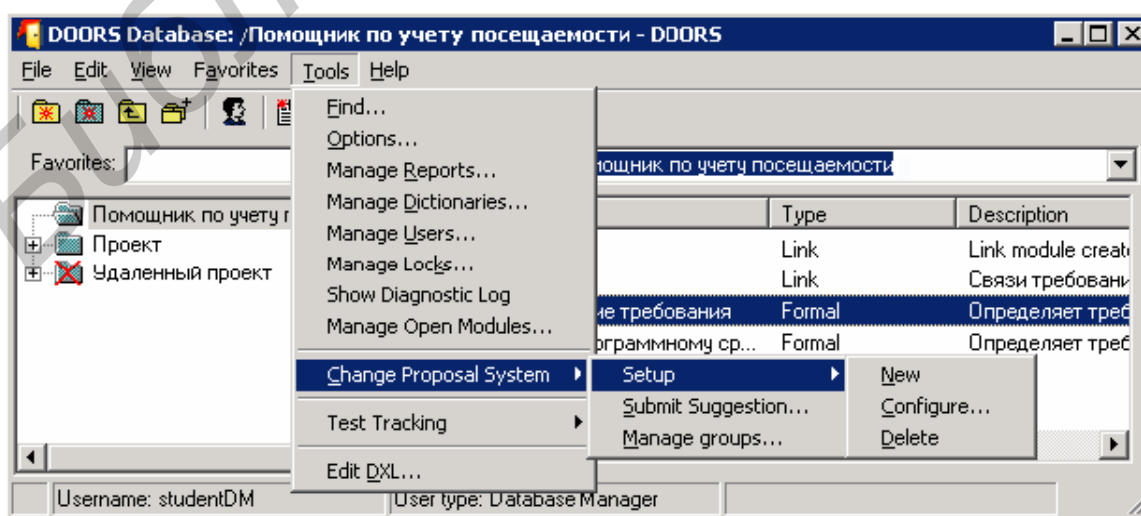


Рис. 2.35. Меню работы с системой предложений о внесении изменений

### 3. ПРИМЕР ЗАДАНИЯ ТРЕБОВАНИЙ В СИСТЕМЕ DOORS

#### 3.1. Постановка задачи

Разработать ПС, предназначенное для учета посещаемости занятий студентами.

#### 3.2. Определение требований

Требования к поставленной задаче задаются в двух формальных модулях:  
– формальный модуль 1: Пользовательские требования;  
– формальный модуль 2: Требования к программному средству (требования разработчика).

Фрагмент иерархии объектов формального модуля пользовательских требований представлен ниже:

1. Информация ввода–вывода.
  - 1.1. Исходные данные.
    - 1.1.1. Название дисциплины.
    - 1.1.2. Номер группы.
    - 1.1.3. Список студентов.
    - 1.1.4. Даты проведения занятий.
    - 1.1.5. Признак присутствия студентов.
  - 1.2. Результирующие данные.
    - 1.2.1. Результаты должны быть представлены в виде таблицы.

...

Фрагмент иерархии объектов формального модуля требований разработчика имеет следующий вид:

1. Представление информации ввода–вывода.
  - 1.1. Представление исходных данных.
    - 1.1.1. Название дисциплины должно выводиться на экран как название таблицы.
    - 1.1.2. Номер группы должен быть выведен как название столбца списка студентов таблицы.
    - 1.1.3. Список студентов выводится в левый столбец таблицы.

...

Для хранения связей требований создается модуль связей «Общие связи». В нем определены следующие наборы связей:

- Пользовательские требования -> Требования к ПС;
- Требования к ПС -> Пользовательские требования;
- Требования к ПС -> Требования к ПС.

Здесь символ -> определяет направление связей, входящих в набор.

Пример набора связей «Пользовательские требования -> Требования к ПС»:

Пользовательское требование 1 -> Требование разработчика 1.  
Пользовательское требование 1.1 -> Требование разработчика 1.1.  
Пользовательское требование 1.1.1 -> Требование разработчика 1.1.1.  
...

## **4. КОНТРОЛЬНЫЕ ВОПРОСЫ**

1. Понятие требований. Уровни требований.
2. Функциональные и нефункциональные требования. Примеры.
3. Основные задачи процесса работы с требованиями.
4. Техники выявления требований.
5. Виды анализа связей требований.
6. Свойства требований. Составные элементы аттестации требований.
7. Понятие управления требованиями. Трассируемость требований.
8. Методы моделирования требований.
9. Краткая характеристика системы DOORS 8.1.
10. Понятие папки, проекта, модуля DOORS. Различия между ними.
11. Формальные модули DOORS. Назначение. Режимы доступа.
12. Представление требований как объектов формальных модулей.

Редактирование объектов.

13. Атрибуты объектов, модулей, связей. Типы атрибутов.
14. Виды в формальном модуле.
15. Поиск, фильтрация и сортировка текстовой информации.
16. Связи требований. Виды, работа со связями требований.
17. Назначение и характеристика модулей связей. Схемы связей объектов.
18. Инструменты анализа связей требований.
19. Базовые версии требований.
20. Цифровые подписи базовых версий.
21. Краткая характеристика системы предложений о внесении изменений.

## **5. ОБЩИЕ ТРЕБОВАНИЯ К ЛАБОРАТОРНОЙ РАБОТЕ**

### **5.1. Порядок выполнения работы**

1. Разработать требования в среде DOORS для заданной преподавателем предметной области. Требования должны содержать не менее трех уровней и должны быть описаны не менее чем в двух формальных модулях (модули пользовательских требований и требований к ПС).
2. Отфильтровать требования по некоторой совокупности их атрибутов.
3. Создать модуль связей и определить в нем связи между требованиями.
4. Провести анализ связей требований с помощью мастера связей.
5. Оформить отчет о выполненной работе.

## **5.2. Возможные последовательности шагов при выполнении работы в среде DOORS 8.1**

### **5.2.1. Работа с объектами требований**

Для создания нового проекта, формального модуля и определения требований в этом модуле следует:

1. Запустить приложение Telelogic DOORS 8.1, воспользовавшись панелью задач «Пуск».

2. Ввести имя пользователя и пароль в окно ввода соответствующих данных (см. рис. 2.1).

3. В проводнике DOORS выбрать папку, в которой необходимо создать формальный модуль (см. рис. 2.4).

4. В главном меню выбрать пункт меню создания нового проекта (см. рис. 2.6).

5. Поместив курсор на созданном проекте, нажать правую кнопку мыши и выбрать пункт создания нового формального модуля.

6. Ввести имя нового формального модуля (см. рис. 2.11).

7. Добавить новый объект требований, выбрав соответствующий пункт меню из контекстного меню формального модуля (см. рис. 2.12).

8. Установить свойства объекта (см. рис. 2.13).

9. Сохранить внесенные изменения.

10. Закрыть приложение DOORS 8.1.

### **5.2.2. Использование функций поиска, фильтрации, сортировки**

Для использования функций поиска, фильтрации и сортировки следует:

1. Запустить приложение Telelogic DOORS 8.1, воспользовавшись панелью задач «Пуск».

2. Ввести имя пользователя и пароль в окно ввода соответствующих данных (см. рис. 2.1).

3. Воспользовавшись проводником DOORS, открыть формальный модуль (см. рис. 2.4).

4. В главном меню формального модуля выбрать пункт Edit > Find (см. рис. 2.9).

5. Ввести информацию о том, что необходимо найти (см. рис. 2.15).

6. Убедиться в верности результатов поиска.

7. Настроить фильтр отображаемой информации, выбрав пункт Tools > Filter главного меню формального модуля (см. рис. 2.19). Применить фильтр, нажав кнопку Apply. Убедиться в верности результатов применения фильтра.

8. Задать порядок сортировки по атрибуту, воспользовавшись пунктом Tools > Sort главного меню формального модуля (см. рис. 2.20). Убедиться в верности результатов сортировки.

### 5.2.3. Работа со связями требований

Для задания связей требований и управления ими следует:

1. Запустить приложение Telelogic DOORS 8.1, воспользовавшись панелью задач «Пуск».
2. Ввести имя пользователя и пароль в окно ввода соответствующих данных (см. рис. 2.1).
3. В проводнике DOORS создать новый модуль связей (см. рис. 2.6).
4. Воспользовавшись проводником DOORS, открыть два формальных модуля (см. рис. 2.9).
5. В первом формальном модуле сделать активным объект требований путем выбора его мышью.
6. В первом формальном модуле выбрать пункт Link > Start Link главного меню (см. рис. 2.22).
7. Во втором формальном модуле выбрать пункт Link > Make Link From Start главного меню (см. рис. 2.22).
8. Выбрать пункт Link > Edit Links меню для того, чтобы увидеть свойства созданной связи требований (см. рис. 2.23).
9. Провести анализ влияния, выбрав пункт Analysis > Impact главного меню формального модуля (см. рис. 2.28).
10. Провести анализ трассируемости, выбрав пункт Analysis > Trace главного меню формального модуля.

### 5.3. Варианты индивидуальных заданий

Самостоятельно определив назначение ПС, в DOORS разработать требования к ним. ПС относятся к следующим предметным областям:

- |                                      |                                   |
|--------------------------------------|-----------------------------------|
| 1) библиотека;                       | 12) заказ и бронирование билетов; |
| 2) отдел кадров;                     | 13) бухгалтерия;                  |
| 3) экзаменационные сессии;           | 14) оплата услуг телефонной сети; |
| 4) общежитие;                        | 15) почтовая служба.              |
| 5) школа;                            | 16) расписание занятий            |
| 6) делопроизводство предприятия;     | преподавателей кафедры;           |
| 7) учет товаров на складах;          | 17) расписание занятий студентов  |
| 8) общественный транспорт;           | университета;                     |
| 9) справочная служба больниц;        | 18) магазины города;              |
| 10) справочная служба аптек;         | 19) домоуправление;               |
| 11) справочная служба бытовых услуг; | 20) банковские услуги.            |



## ЛИТЕРАТУРА

1. IEEE 830-1998 – IEEE Recommended Practice for Software Requirements Specifications.
2. Бахтизин, В. В. Методология структурного анализа и проектирования SADT : учеб. пособие по курсу «Технология проектирования программ» для студ. спец. Т10.02.00 / В. В. Бахтизин, Л. А. Глухова. – Минск : БГУИР, 2001.
3. Бахтизин, В. В. Структурный анализ и моделирование в среде CASE-средства VPwin : учеб. пособие по курсу «Технология проектирования программ» для студ. спец. 40 01 01 «Программное обеспечение информационных технологий» / В. В. Бахтизин, Л. А. Глухова. – Минск : БГУИР, 2002.
4. Вигерс, К. Разработка требований к программному обеспечению / К. Вигерс. – М. : Русская Редакция, 2004.
5. Hull, E. Requirements engineering / E. Hull, K. Jackson, J. Dick. – Springer, 2005.
6. Kotonya, G. Requirements Engineering. Processes and techniques / G. Kotonya, I. Sommerville. – John Wiley & Sons, 1998.
7. Leffingwell, D. Managing Software Requirements / D. Leffingwell, D. Widrig. – Addison Wesley, 1999.

Учебное издание

**Бахтизин Вячеслав Вениаминович**  
**Глухова Лилия Александровна**  
**Неборский Сергей Николаевич**

***РАБОТА С ТРЕБОВАНИЯМИ***  
***В СРЕДЕ DOORS***

УЧЕБНО-МЕТОДИЧЕСКОЕ ПОСОБИЕ

Редактор *Н. В. Гриневич*  
Корректор *М. В. Тезина*  
Компьютерная верстка *Е. Г. Реут*

Подписано в печать 29.05.2007. Формат 60x84 1/16. Бумага офсетная. Гарнитура «Таймс».  
Печать ризографическая. Усл. печ. л. 2,44. Уч.-изд. л. 2,0. Тираж 100 экз. Заказ 137.

Издатель и полиграфическое исполнение: Учреждение образования  
«Белорусский государственный университет информатики и радиоэлектроники»  
ЛИ №02330/0056964 от 01.04.2004. ЛП №02330/0131666 от 30.04.2004.  
220013, Минск, П. Бровки, 6