

12_Этапы проектирования ИС с применением UML

Основные типы UML-диаграмм, используемые в проектировании информационных систем. Взаимосвязи между диаграммами. Поддержка UML итеративного процесса проектирования ИС. Этапы проектирования ИС: моделирование бизнес-прецедентов, разработка модели бизнес-объектов, разработка концептуальной модели данных, разработка требований к системе, анализ требований и предварительное проектирование системы, разработка моделей базы данных и приложений, проектирование физической реализации системы.

UML обеспечивает поддержку всех этапов жизненного цикла ИС и предоставляет для этих целей ряд графических средств – диаграмм.

На этапе создания *концептуальной модели* для описания бизнес-деятельности используются *модели бизнес-прецедентов* и диаграммы видов деятельности, для описания бизнес-объектов – *модели бизнес-объектов* и диаграммы последовательностей.

На этапе создания логической модели ИС описание требований к системе задается в виде модели и описания системных прецедентов, а предварительное проектирование осуществляется с использованием диаграмм классов, диаграмм последовательностей и диаграмм состояний.

На этапе создания физической модели детальное проектирование выполняется с использованием диаграмм классов, диаграмм компонентов, диаграмм развертывания.

Ниже приводятся определения и описывается назначение перечисленных диаграмм и моделей применительно к задачам проектирования ИС (в скобках приведены альтернативные названия диаграмм, использующиеся в современной литературе).

Диаграммы прецедентов (диаграммы вариантов использования, use case diagrams) – это обобщенная модель функционирования системы в окружающей среде.

Диаграммы видов деятельности (диаграммы деятельности, activity diagrams) – модель бизнес-процесса или поведения системы в рамках прецедента.

Диаграммы взаимодействия (interaction diagrams) – модель процесса обмена сообщениями между объектами, представляется в виде диаграмм последовательностей (sequence diagrams) или кооперативных диаграмм (collaboration diagrams).

Диаграммы состояний (statechart diagrams) – модель динамического поведения системы и ее компонентов при переходе из одного состояния в другое.

Диаграммы классов (class diagrams) – логическая модель базовой структуры системы, отражает статическую структуру системы и связи между ее элементами.

Диаграммы базы данных (database diagrams) — модель структуры базы данных, отображает таблицы, столбцы, ограничения и т.п.

Диаграммы компонентов (component diagrams) – модель иерархии подсистем, отражает физическое размещение баз данных, приложений и интерфейсов ИС.

Диаграммы развертывания (диаграммы размещения, deployment diagrams) – модель физической архитектуры системы, отображает аппаратную конфигурацию ИС.

На [рис. 12.1](#) показаны отношения между различными видами диаграмм UML. Указатели стрелок можно интерпретировать как отношение "является источником входных данных для..." (например, диаграмма прецедентов является источником данных для диаграмм видов деятельности и последовательности). Приведенная схема является наглядной иллюстрацией итеративного характера разработки моделей с использованием UML.

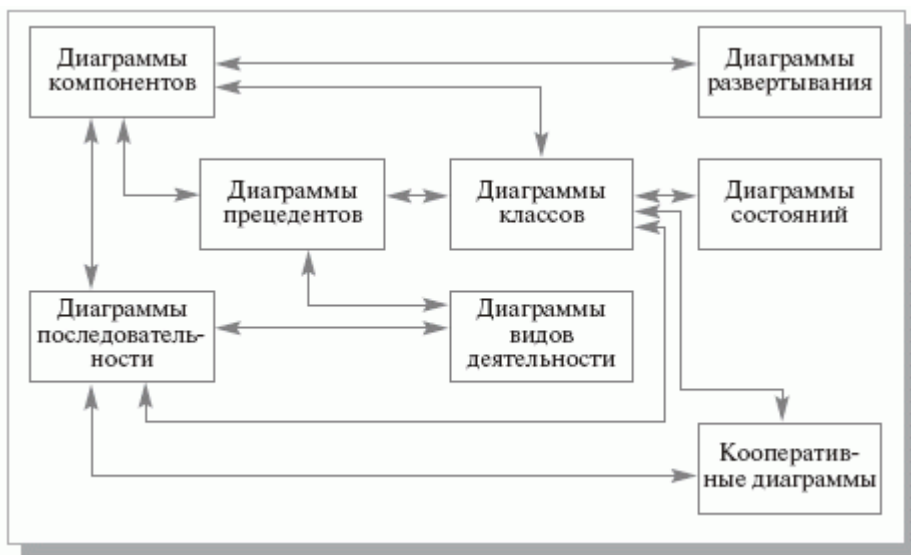


Рис. 12.1. Взаимосвязи между диаграммами UML

Ниже приводятся описания последовательных этапов проектирования ИС с использованием UML.

Разработка модели бизнес-прецедентов

Модель бизнес-прецедентов описывает бизнес-процессы с точки зрения внешнего пользователя, т.е. отражает взгляд на деятельность организации извне.

Проектирование системы начинается с изучения и моделирования бизнес-деятельности организации. На этом этапе вводится и отображается в модели ряд понятий, свойственных объектно-ориентированному подходу:

Исполнитель (Действующее лицо, Актор) – личность, организация или система, взаимодействующая с ИС; различают внешнего исполнителя (который использует или используется системой, т.е. порождает прецеденты деятельности) и внутреннего исполнителя (который обеспечивает реализацию прецедентов деятельности внутри системы). На диаграмме исполнитель представляется стилизованной фигуркой человека.

Прецедент – законченная последовательность действий, инициированная внешним объектом (личностью или системой), которая взаимодействует с ИС и получает в результате некоторое сообщение от ИС. На диаграмме представляется овалом с надписью, отражающей содержание действия.

Класс — описание совокупности однородных объектов с их атрибутами, операциями, отношениями и семантикой. На диаграмме представляется прямоугольником, содержащим описание атрибутов и операций класса.

Ассоциация – связь между двумя элементами модели. На диаграмме представляется линией.

Обобщение – связь между двумя элементами модели, когда один элемент (подкласс) является частным случаем другого элемента (суперкласса). На диаграмме представляется стрелкой.

Агрегация – отношение между элементами модели, когда один элемент является частью другого элемента (агрегата). На диаграмме представляется стрелкой с ромбовидным концом.

Для иллюстрации этапов разработки проекта использованы адаптированные материалы проекта ИС медицинского центра [рис. 12.2]. Назначение ИС – автоматизация ведения и использования клинических записей о пациентах. В настоящее время эта работа производится вручную персоналом центра. На рис. 12.2 представлена общая модель деятельности центра в виде диаграммы прецедентов. Прецедент "Обслуживание пациента" реализуется через множество других, более ограниченных прецедентов (рис. 12.3), отражающих детализацию представления функционирования центра.



Рис. 12.2. Общая диаграмма деятельности медицинского центра по обслуживанию пациента

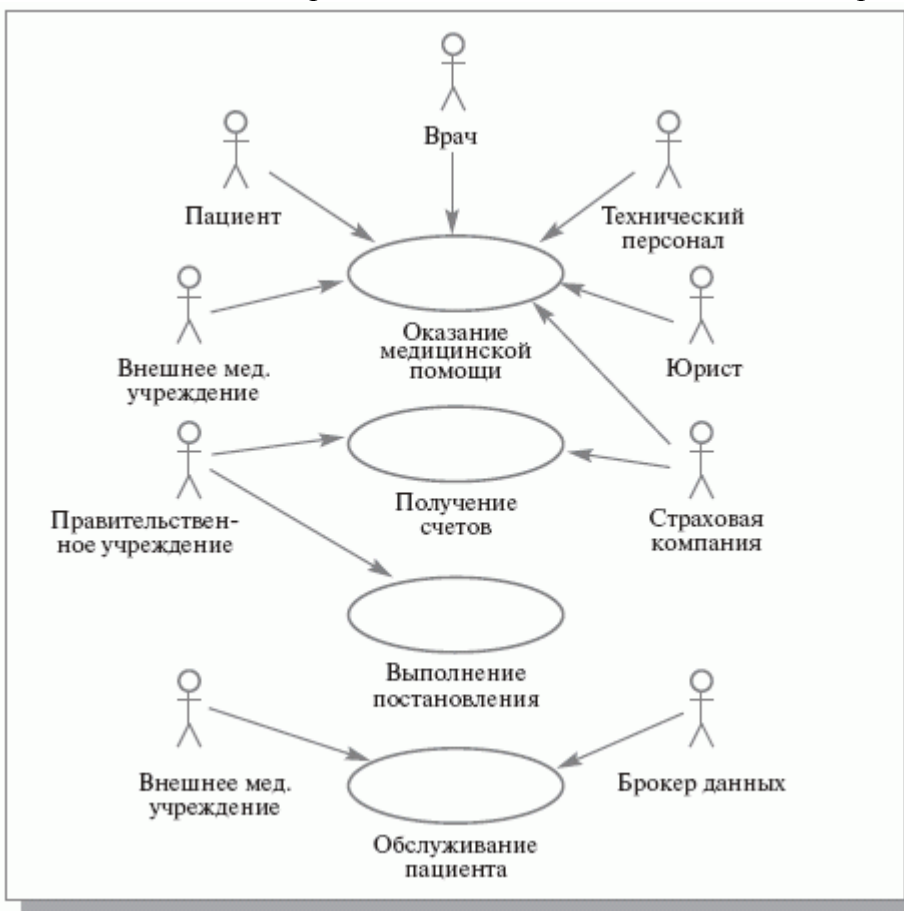


Рис. 12.3. Модель бизнес-прецедентов, составляющих обслуживание пациента
Для включения в диаграмму выбранные прецеденты должны удовлетворять следующим критериям:

- прецедент должен описывать, **ЧТО** нужно делать, а не **КАК**;
- прецедент должен описывать действия с точки зрения **ИСПОЛНИТЕЛЯ**;
- прецедент должен возвращать исполнителю некоторое **СООБЩЕНИЕ**;

- последовательность действий внутри прецедента должна представлять собой одну **НЕДЕЛИМУЮ** цепочку.

Исходя из цели создания системы, для дальнейшего исследования и моделирования отбираются только те бизнес-прецеденты, которые связаны с использованием клинических записей.

Выполнение прецедента описывается с помощью диаграмм видов деятельности, которые отображают исполнителей и последовательность выполнения соответствующих бизнес-процессов ([рис. 12.4](#)).

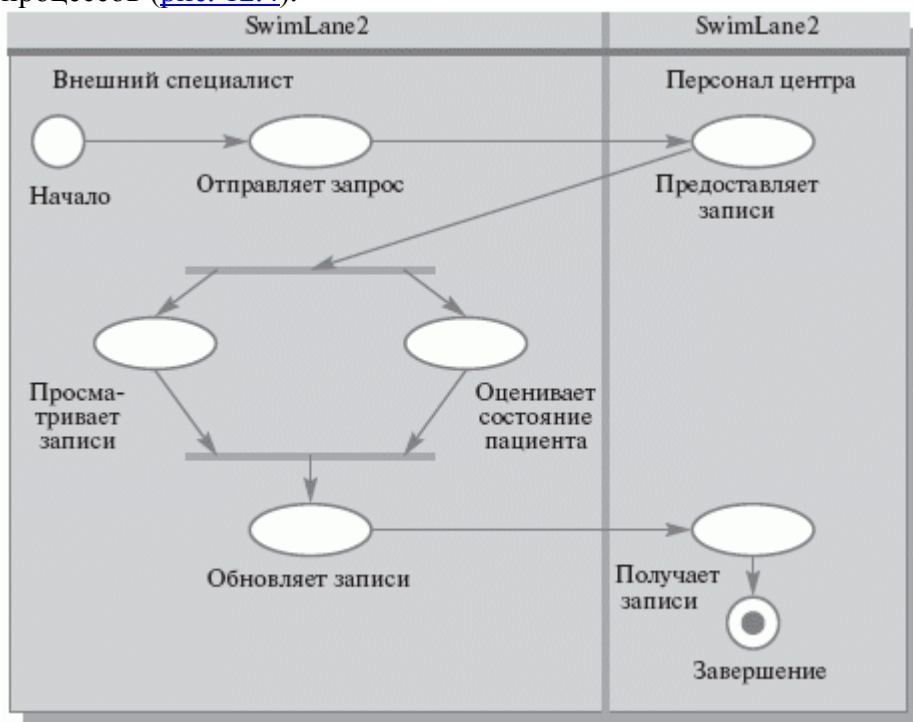


Рис. 12.4. Диаграмма видов деятельности для прецедента "Оказание медицинской помощи" Несмотря на то, что оказание медицинской помощи предусматривает множество разнообразных действий исполнителей, для нашей задачи существенными являются только процессы обмена информацией между этими исполнителями, и именно они отображаются в создаваемых моделях. Поэтому на диаграмме отражен процесс оценки состояния пациента на основании имеющейся в центре информации о нем.

Общее поле диаграммы деятельности делится на несколько "плавательных дорожек", каждая из которых содержит описание действий одного из исполнителей. Основными элементами диаграмм видов деятельности являются обозначения состояния ("начало", "конец"), действия (овал) и момента синхронизации действий (линейка синхронизации, на которой сходятся или разветвляются несколько стрелок).

Диаграмма подходит для описания действий как внешнего, так и внутреннего специалиста центра.

Этап завершается после разработки диаграмм видов деятельности для всех выделенных в модели бизнес-прецедентов. Естественно, на последующих этапах анализа и проектирования будут выявлены какие-то важные подробности в описании деятельности объекта автоматизации. Поэтому разработанные на данном этапе модели будут еще неоднократно корректироваться.

Разработка модели бизнес-объектов

Следующим этапом проектирования ИС является разработка модели бизнес-объектов, которая показывает выполнение бизнес-процессов организации ее внутренними исполнителями. Основными компонентами моделей бизнес-объектов являются внешние и внутренние исполнители, а также бизнес-сущности, отображающие все, что используют внутренние исполнители для реализации бизнес-процессов. Пример модели бизнес-объектов для прецедента "Ответ на запрос" приведен на [рис. 12.5](#).



Рис. 12.5. Модель бизнес-объектов прецедента "Ответ на запрос"

В этой диаграмме появилось новое действующее лицо – отправитель запроса. На самом деле с запросом о состоянии пациента могут обращаться в систему многие из действующих лиц: юрист, страховая компания, технический персонал и даже сам пациент. Таким образом, понятие "Отправитель запроса" служит для обобщенного представления всех этих действующих лиц при описании прецедента "Ответ на запрос" (рис. 12.6). "Отправитель запроса" становится суперклассом по отношению к обобщаемым понятиям (подклассам).



Рис. 12.6. Обобщение классов

Для детального описания выполнения бизнес-процессов обычно используются диаграммы последовательностей (рис. 12.7).



Рис. 12.7. Диаграмма последовательностей для прецедента "Ответ на запрос"

Основными элементами диаграммы последовательностей являются обозначения объектов (прямоугольники), вертикальные линии, отображающие течение времени при деятельности объекта, и стрелки, показывающие выполнение действий объектами.

Результатом этого этапа являются согласованные с заказчиком и достаточно подробные описания действий специалистов организации, внедряющей ИС, необходимые для обеспечения исполнения ее функций.

Разработка концептуальной модели данных

Затем на основе информации, выявленной на этапах бизнес-моделирования, выполняется разработка *концептуальной модели данных*, которые будут использоваться в разрабатываемой системе. На рис. 12.8 представлена в виде диаграммы классов модель данных для объекта "Клинические записи".

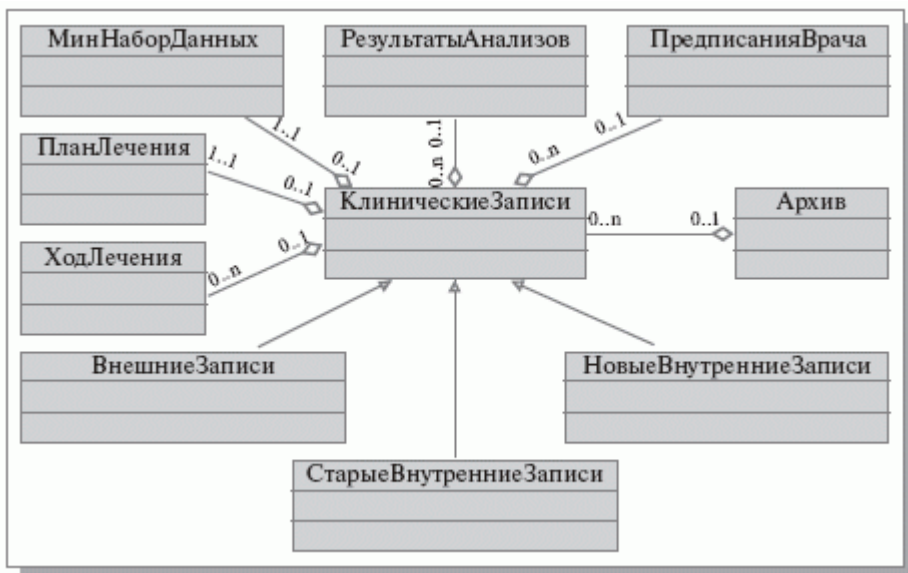


Рис. 12.8. Концептуальная модель данных

Модель показывает, что клинические записи включают (агрегируют) ряд блоков. При этом "минимальный набор данных" и "план лечения" могут быть включены в каждую клиническую запись в единственном экземпляре, а блоки "результаты анализов", "предписания врача", "ход лечения" могут повторяться неограниченное число раз. Архив состоит из множества клинических записей (агрегирует клинические записи), но может быть и пустым.

Поскольку пациент может предварительно проходить лечение в других учреждениях, или несколько раз проходить лечение в центре, появляются дополнительные разновидности (подклассы) клинических записей: внешние, старые внутренние, новые внутренние. Этот этап завершает процедуры бизнес-моделирования и позволяет представить команде проектировщиков в едином формате ту информацию, которая будет необходима для создания системы. Разработанные диаграммы являются отправной точкой в процессах проектирования баз данных и приложений системы, обеспечивают согласованность действий бизнес-аналитиков и разработчиков в процессе дальнейшей работы над системой. Эти диаграммы, конечно же, будут претерпевать изменения в процессе последующего проектирования, однако эти изменения будут фиксироваться в формате, уже привычном для всей команды разработчиков, и будут автоматически отражаться в последующих моделях.

Разработка требований к системе

На этапе формирования требований, прежде всего, необходимо определить область действия разрабатываемой системы и получить точное представление о желаемых возможностях системы.

Основой разработки требований является *модель системных прецедентов*, отражающая выполнение конкретных обязанностей внутренними и внешними исполнителями с использованием ИС.

Источником данных для создания *модели системных прецедентов* являются разработанные на предыдущем этапе бизнес-модели. Однако при создании модели полезно предварительно составить детальные описания прецедентов, содержащие определения используемых данных и точную последовательность их выполнения. Описание осуществляется в соответствии с принятым в организации шаблоном, который обычно включает следующие разделы:

- заголовок (название прецедента, ответственный за исполнение, дата создания шаблона/внесения изменений);
- краткое описание прецедента;
- ограничения;
- предусловия (необходимое состояние системы или условия, при которых должен выполняться прецедент);
- постусловия (возможные состояния системы после выполнения прецедента);

- предположения;
- основная последовательность действий;
- альтернативные последовательности действий и условия, их инициирующие;
- точки расширения и включения прецедентов.

В процессе создания *модели системных прецедентов* осуществляется преобразование и перенос компонентов бизнес-моделей на новые диаграммы. Типовые преобразования по технологии Rational Unified Process приведены в [таблица 12.1](#).

Таблица 12.1.

Элементы бизнес-модели	Элементы модели системных прецедентов
Бизнес-прецеденты	Подсистемы
Внешние исполнители	Исполнители
Внутренние исполнители	Исполнители или прецеденты
Процессы, выполняемые внутренними исполнителями	Прецеденты

На [рис. 12.9](#) представлена *модель системных прецедентов* для бизнес-прецедента "Оказание медицинской помощи". Исходя из цели создания системы, в *модели системных прецедентов* отражены только те действия исполнителей, которые связаны с предоставлением доступа и обновлением клинических записей.

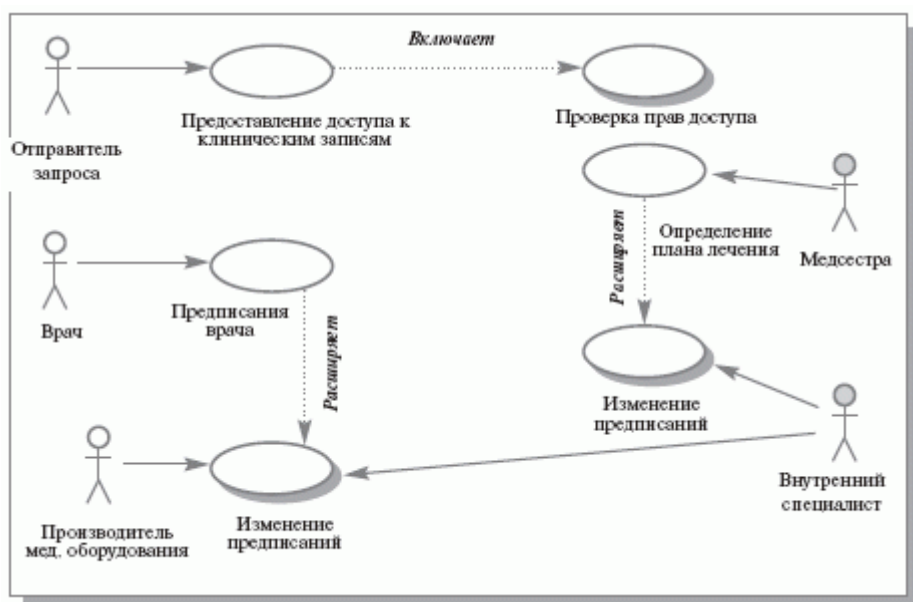


Рис. 12.9. Модель системных прецедентов

Описываемые моделью функции характерны только для одного вида деятельности – оказания медицинской помощи, и в основном не используются в других видах деятельности Центра. Это позволяет объединить выделенные функции в некую единую подсистему проектируемой ИС.

Внутренний исполнитель "Персонал центра" (см. [рис. 12.4](#), [рис. 12.7](#)) и выполняемый им ручной процесс преобразован в системный прецедент "Предоставление доступа к клиническим записям".

Внешние исполнители (например, "Производитель медицинского оборудования") непосредственно взаимодействуют с проектируемой системой, т.е. превращаются в исполнителей.

В модели отражены два специальных типа связи между прецедентами (на [рис. 12.9](#) соответствующие прецеденты выделены тенью):

- "включает" — один прецедент в процессе своего исполнения обязательно выполняет некий блок действий, составляющих другой прецедент;

- "расширяет" — когда прецеденты подобны по своим действиям, но один несет несколько большую функциональную нагрузку.

Прецедент "Проверка прав доступа" впервые появился на диаграммах и реализуется средствами разрабатываемой ИС. Поэтому для него приходится разрабатывать диаграмму последовательностей, описывающую его исполнение (рис. 12.10). В результате в проектируемой ИС появляются два новых объекта – программный модуль "Менеджер защиты" и информационный блок "Набор прав".

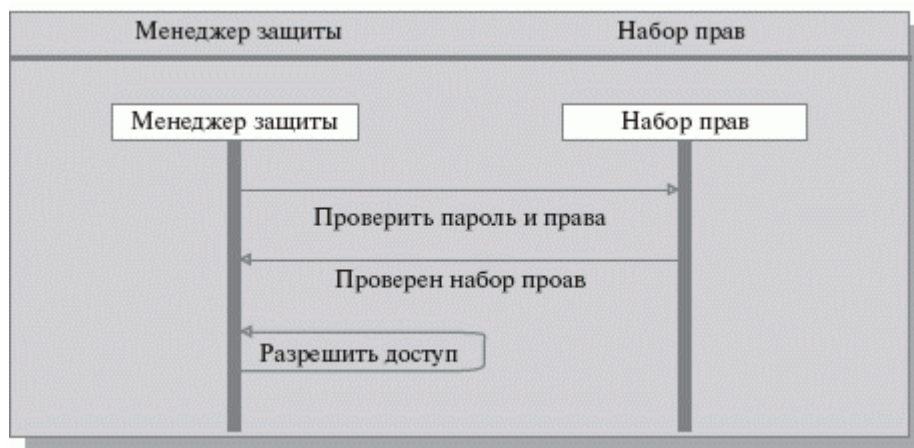


Рис. 12.10. Диаграмма последовательностей для прецедента "Проверка прав"

Таким образом, результатом разработки *модели системных прецедентов* является не только исчерпывающий перечень функций, которые должны быть реализованы в проектируемой системе, но и подробное описание необходимой реализации этих функций.

Анализ требований и предварительное проектирование системы.

Основные задачи этапа:

- определить проект системы, который будет отвечать всем бизнес-требованиям;
- разработать общий предварительный проект для всех команд разработчиков (проектировщиков баз данных, разработчиков приложений, системных архитекторов и пр.)

Основным инструментом на данном этапе являются диаграммы классов системы, которые строятся на основе разработанной *модели системных прецедентов*. Одновременно на этом этапе уточняются диаграммы последовательностей выполнения отдельных прецедентов, что приводит к изменениям в составе объектов и диаграммах классов. Это естественное отражение средствами UML итеративного процесса разработки системы.

Диаграммы классов системы заполняются объектами из *модели системных прецедентов*. Они содержат описание этих объектов в виде классов и описание взаимодействия между классами.

Диаграмма классов, описывающая процедуры защиты доступа к данным, приведена на [рис. 12.11](#).

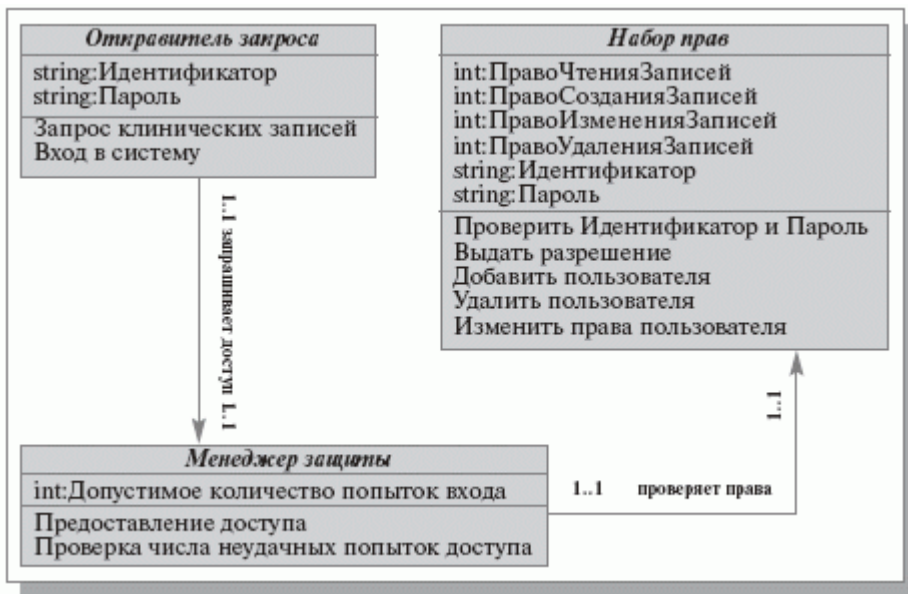


Рис. 12.11. Диаграмма классов "Защита доступа"

Таким образом, в результате этого этапа проектирования появляется достаточно подробное описание состава и функций проектируемой системы, а также информации, которую необходимо использовать в базе данных и в приложениях.

Поскольку диаграммы классов строятся на основе разработанных ранее бизнес-моделей, появляется уверенность в том, что разрабатываемая система будет действительно удовлетворять исходным требованиям заказчика.

В то же время, благодаря своему синтаксису, диаграммы классов оказываются хорошим средством структурирования и представления требований к функциональности, интерфейсам и данным для элементов проектируемой системы.

Разработка моделей базы данных и приложений

На этом этапе осуществляется отображение элементов полученных ранее моделей классов в элементы моделей базы данных и приложений:

- классы отображаются в таблицы;
- атрибуты – в столбцы;
- типы – в типы данных используемой СУБД;
- ассоциации – в связи между таблицами (ассоциации "многие-ко-многим" преобразуются в ассоциации "один-ко-многим" посредством создания дополнительных таблиц связей);
- приложения – в отдельные классы с окончательно определенными и связанными с данными в базе методами и атрибутами.

Поскольку модели базы данных и приложений строятся на основе единой логической модели, автоматически обеспечивается связность этих проектов ([рис. 12.12](#)).

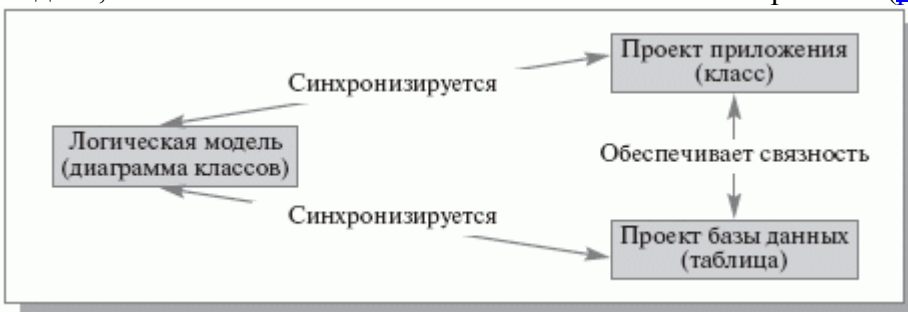


Рис. 12.12. Связь между проектами базы данных и приложений

В модель базы данных отображаются только перманентные классы, из которых удаляются атрибуты, не отображаемые в столбцах (например, атрибут типа "Общий объем продаж", который получается суммированием содержимого множества полей базы данных).

Некоторые атрибуты (например, АДРЕС) могут отображаться в множество столбцов (СТРАНА, ГОРОД, УЛИЦА, ДОМ, ПОЧТОВЫЙ ИНДЕКС).

Для каждого простого класса в модели базы данных формируется отдельная таблица, включающая столбцы, соответствующие атрибутам класса.

Отображение классов подтипов в таблицы осуществляется одним из стандартных способов:

- одна таблица на класс;
- одна таблица на суперкласс;
- одна таблица на иерархию.

В первом случае для каждого из классов создается отдельная таблица, между которыми затем устанавливаются необходимые связи. Во втором случае создается таблица для суперкласса, а затем в каждую таблицу подклассов включаются столбцы для каждого из атрибутов суперкласса. В третьем – создается единая таблица, содержащая атрибуты как суперкласса, так и всех подклассов (рис. 12.13). При этом для выделения исходных таблиц подклассов в результирующую таблицу добавляется один или более дополнительных столбцов (на рисунке показан курсивом).

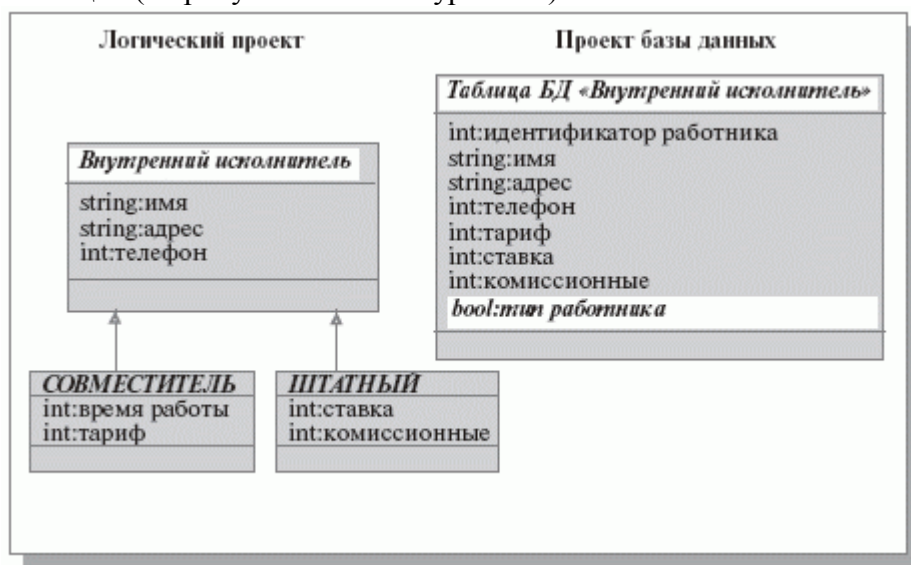


Рис. 12.13. Преобразование иерархии в таблицу

Разработка проекта базы данных осуществляется с использованием специального UML-профиля (Profile for Database Design), который включает следующие основные компоненты диаграмм:

- таблица – набор записей базы данных по определенному объекту;
- столбец – элемент таблицы, содержащий значения одного из атрибутов таблицы;
- первичный ключ (PK) – атрибут, однозначно идентифицирующий строку таблицы;
- внешний ключ (FK) – один или группа атрибутов одной таблицы, которые могут использоваться как первичный ключ другой таблицы;
- обязательная связь – связь между двумя таблицами, при которой дочерняя таблица существует только вместе с родительской;
- необязательная связь – связь между таблицами, при которой каждая из таблиц может существовать независимо от другой;
- представление – виртуальная таблица, которая обладает всеми свойствами обычной таблицы, но не хранится постоянно в базе данных;
- хранимая процедура – функция обработки данных, выполняемая на сервере;
- домен – множество допустимых значений для столбца таблицы.

На рис. 12.14 представлен фрагмент модели базы данных — две таблицы, соответствующие классам "пациент" (рис. 12.3, рис. 12.6) и "минимальный набор данных" (рис. 12.8). Связь между ними обязательная, поскольку "минимальный набор данных" не может существовать без "пациента".

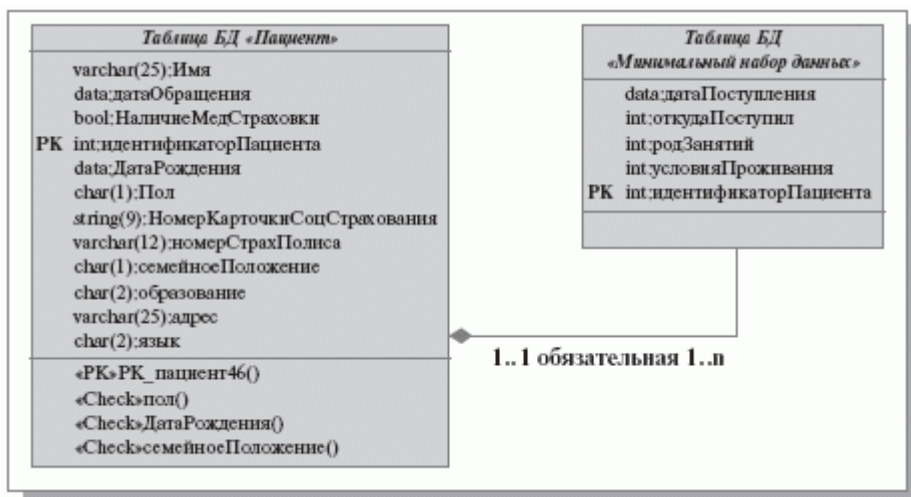


Рис. 12.14. Фрагмент модели базы данных

На диаграммах указываются дополнительные характеристики таблиц и столбцов:

- ограничения – определяют допустимые значения данных в столбце или операции над данными (ключ (PK,FK) – ограничение, определяющее тип ключа и его столбец; проверка (Check) – ограничение, определяющее правило контроля данных; уникальность (Unique) – ограничение, определяющее, что в столбце содержатся неповторяющиеся данные);
- триггер – программа, выполняющая при определенных условиях предписанные действия с базой данных;
- тип данных и пр.

Результатом этапа является детальное описание проекта базы данных и приложений системы.

Проектирование физической реализации системы

На этом этапе проектирования модели баз данных и приложений дополняются обозначениями их размещения на технических средствах разрабатываемой системы. На [рис. 12.15](#) приведено изображение разделения таблицы "пациент" на три экстенда (<<Tablespace>>) в соответствии с первой буквой фамилии пациента.

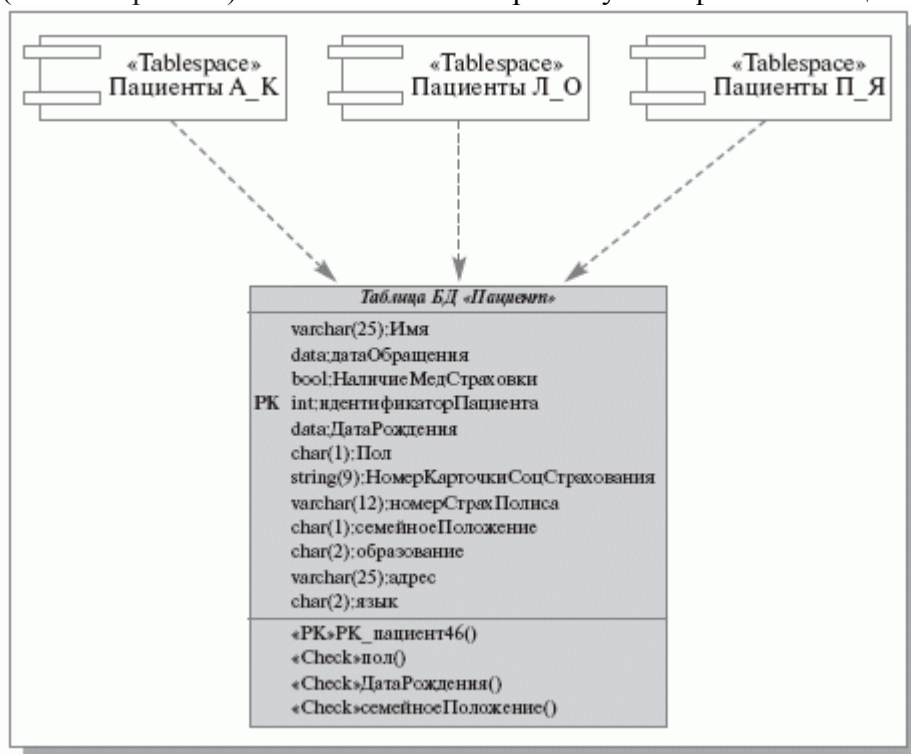


Рис. 12.15. Экстенды таблицы "Пациент"

Основными понятиями UML, которые используются на данном этапе, являются следующие:

- компонент – самостоятельный физический модуль системы;
- зависимость – связь между двумя элементами, при которой изменения в одном элементе вызывают изменения другого элемента;
- устройство – узел, не обрабатывающий данные;
- процессор – узел, выполняющий обработку данных;
- соединение – связь между устройствами и процессорами.

Диаграммы развертывания позволяют отобразить на единой схеме различные компоненты системы (программные и информационные) и их распределение по комплексу технических средств (рис. 12.16).

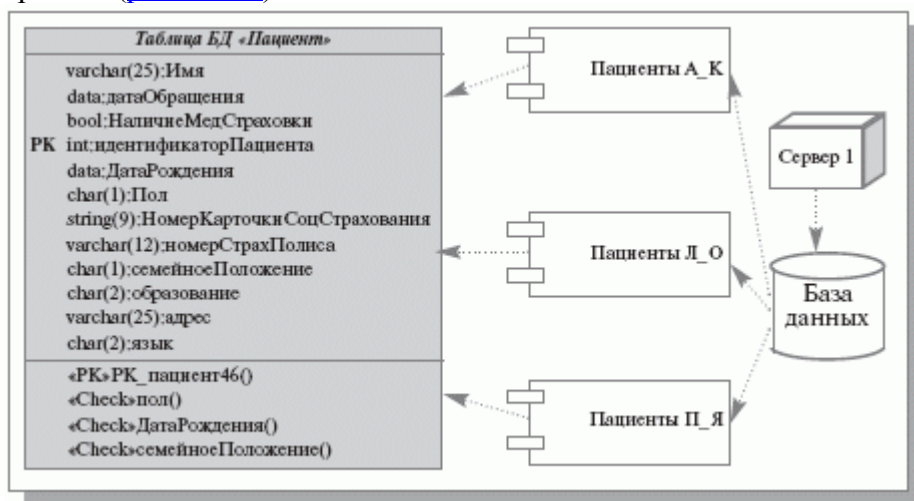


Рис. 12.16. Фрагмент диаграммы развертывания ИС

Таким образом, при проектировании сложной ИС она разделяется на части, и каждая из них затем исследуется и создается отдельно. В настоящее время используются два различных способа такого разбиения ИС на подсистемы: структурное (или функциональное) разбиение и объектная (компонентная) декомпозиция.

С позиций проектирования ИС суть функционального разбиения может быть выражена известной формулой: "Программа = Данные + Алгоритмы". При функциональной декомпозиции программной системы ее структура описывается блок-схемами, узлы которых представляют собой "обрабатывающие центры" (функции), а связи между узлами описывают движение данных.

При объектном разбиении в системе выделяются "активные сущности" – объекты (или компоненты), которые взаимодействуют друг с другом, обмениваясь сообщениями и выполняя соответствующие функции (методы) объекта.

Если при проектировании ИС разбивается на объекты, то для ее визуального моделирования следует использовать UML. Если в основу проектирования положена функциональная декомпозиция ИС, то UML не нужен и следует использовать рассмотренные ранее структурные нотации.

В то же время, при выборе подхода к разработке ИС следует учитывать, что визуальные модели все более широко используются в существующих технологиях управления проектированием систем, сложность, масштабы и функциональность которых постоянно возрастают. Они хорошо приспособлены для решения таких часто возникающих при создании систем задач как: физическое перераспределение вычислений и данных, обеспечение параллелизма вычислений, репликация БД, обеспечение безопасности доступа к ИС, оптимизация балансировки нагрузки ИС, устойчивость к сбоям и т.п.

Визуализированные средствами UML модели ИС позволяют наладить плодотворное взаимодействие между заказчиками, пользователями и командой разработчиков. Они обеспечивают ясность представления выбранных архитектурных решений и позволяют понять разрабатываемую систему во всей ее полноте.

