

**Олег Викторович Рогозин**, доцент МГТУ им. Баумана,  
кандидат технических наук

## ТЕОРИЯ И ПРАКТИКА ПРОЕКТИРОВАНИЯ

*Статья посвящена разработке модели представления знаний в Web-приложении, демонстрирующей работу сервера дистанционного обучения, соответствующего международным стандартам в этой области. SCORM-технология включает использование модели агрегации контента (SAM) и среду для выполнения обучающих объектов. Разработанная модель реализована в виде Web-приложения.*

Обучение через Web — интенсивная область исследований и разработок. Польза от использования Web для обучения основана на возможности размещения обучаемых на любых удалённых расстояниях, при этом процесс обучения сохраняет множество отличительных черт, свойственных обучению «живую». Приложение, установленное и поддерживаемое в одном месте, может использоваться обучаемыми по всему миру, имеющими компьютер с любым видом подключения к Интернету. Множество курсов и других обучающих приложений стали доступны в Web за последние годы

Основной проблемой дистанционного образования по-прежнему остаётся недостаток качественного методического обеспечения. В 1980–90-е годы разрабатывалось большое количество компьютерных обучающих

программ, но их смысловая информация (контент) была неразрывно связана со средой хранения-доставки-публикации. Устаревание среды и формата хранения привело к невозможности дальнейшего использования разработанных тестов и курсов.

В связи с этим существует потребность в новых разработках таких систем дистанционного обучения, отличительными особенностями которых являются универсальность используемых информационных обучающих ресурсов, их модульность; возможность работы на различных аппаратно-программных платформах, в сети (в том числе локальной), возможности масштабирования системы для использования в учебных учреждениях различной структуры и величины, с различными уровнями и целями использования технологий дистанционного обучения; обеспечение высокого уровня надёжности системы; полная автоматизация функционирования системы и возможность интеграции с другими видами информационных систем.

### Основные модели SCORM-технологии

SCORM представляет собой взаимосвязанный набор спецификаций и стандартов. Почти все они были разработаны ранее в других организациях (в частности, в таких орга-

низациях, как IMS, AICC, ARIADNE и IEEE LTSC) и затем обобщены в SCORM. На данный момент все составляющие стандарты SCORM сгруппированы в три категории: модель агрегирования учебных материалов или контента (Content Aggregation Model — CAM), среда времени выполнения (Run-time Environment — RTE), упорядочивание и навигация (Sequencing and Navigation — SN)

#### **Модель агрегации контента (Content Aggregation Model — CAM)**

Модель описывает компоненты, используемые при обучении, создание из них упаковок для обмена между информационными системами, способы описания компонентов для дальнейшего поиска, а также способы создания правил упорядочивания. Упаковка контента (Content Package) представляет собой структурированный набор объектов контента (Content Objects), описанных в файле-манифесте. Упаковка контента может представлять собой учебный курс, лекцию, модуль или просто набор связанных объектов контента. Манифест является обязательной частью любой упаковки контента и представляет собой XML файл с именем «imsmanifest.xml». В файле описывается содержимое упаковки, может содержаться дополнительное описание структуры упаковки.

#### **Модель агрегации контента состоит из компонентов:**

- Модель контента (компоненты контента, используемые в обучении).
- Упаковка контента (структура контента и процесс упаковки).
- Метаданные (описание компонентов модели контента).
- Упорядочивание и навигация (набор правил для упорядочивания).

#### **Модель Среды Времени Выполнения (Run-time Environment — RTE)**

Модель Среды Времени Выполнения определяет требования, предъявляемые к объектам контента, использованию ими API и модели данных RTE.

Задача RTE — предоставить средства взаимодействия между объектами контента и системой управления обучением (LMS). SCORM обеспечивает независимость учебного контента от LMS. Это обеспечивается следующими компонентами RTE:

- Запуск (Launch).
- API (взаимодействие с LMS).
- Модель данных (Data Model).

API определяет заданный заранее набор выполняемых функций для взаимодействия между LMS и запускаемыми им объектами контента. Процесс запуска объекта контента завершается после «рукопожатия» (handshake) между объектом контента и LMS. После того, как объект становится ненужным, «рукопожатие» разрывается. Также API позволяет читать/записывать данные LMS и производить контроль ошибок.

Модель данных RTE предоставляет «словарь» для обмена информацией с LMS (для чтения и записи данных при вызове API функций).

#### **Модель упорядочивания и управления (Sequencing and Navigation — SN)**

Модель описывает возможность упорядочивания SCORM-совместимого контента при помощи серии навигационных событий (navigation events), инициированных пользователем или системой. Ветвление и поток контента могут быть описаны заданным заранее (обычно при создании) набором действий. В модели также описывается интерпре-

тация SCORM-совместимой LMS правил упорядочения, заданных разработчиком контента, а также набора инициированных учащимся или системой навигационных событий и их влияния на среду времени выполнения. SN-модель представляет ветвление и поток обучающих действий в виде Дерева Действий (Activity Tree), основанного на результатах взаимодействия учащегося объектами контента и стратегии упорядочивания, заданной разработчиком. Дерево Действий — концептуальная структура действий учащегося, создаваемая LMS для каждого учащегося. В модели SCORM действие учащегося может ссылаться на объект контента, который ему доставлен.

Модель SN описывает способы приведения в действие и обработки навигационных событий, инициированных учащимся или системой. Каждое обучающее действие имеет связанный с ним объект контента. RTE описывает способ запуска объектов контента. Последовательность представленных объектов контента для данных учащегося и структуры контента представляет собой обучающий опыт (learning experience). Модель RTE также описывает обработку LMS полученного обучающего опыта и то, как этот опыт может повлиять на Дерево Действий.

Многие понятия из модели SAM связаны с моделью SN. В частности, SAM описывает создание правил упорядочивания и записи их на XML.

### Структура учебного курса. Описание и упаковка учебных материалов

Выбранный формат для хранения учебных курсов соответствует основным требова-

ниям международного стандарта IMS Content Packaging. Спецификация описывает способы описания и упаковки учебных материалов, например, отдельных курсов или их коллекции, в интероперабельные (легко распределяемые) блоки за счёт описания состава, структуры и расположения предоставляемых учебных материалов с указанием типа содержимого. Стандарт рассчитан на разработчиков учебных материалов, продавцов обучающих систем, компьютерных платформ и провайдеров обучающих сервисов. Обучающие материалы описываются и упаковываются с использованием XML-формата, что позволяет добиться интероперабельности с любыми приложениями и инструментами, поддерживающими эту спецификацию.

В соответствии с этим стандартом учебный курс представляется в виде некоего независимого «пакета» (рис. 2.1), внутри которого содержатся непосредственно содержимое (ресурсы) и манифест (описание). В качестве ресурсов в «пакете» могут содержаться самые разнообразные виды файлов — начиная от аудио- и видеоформатов и заканчивая HTML-страницами и простыми текстовыми файлами.

Манифест представляет собой описание внутреннего содержимого в виде XML файла, имеющего фиксированное имя **imsmanifest.xml** и определённый формат<sup>1</sup>. Манифест описывает метаданные, информацию об организации учебных материалов, описание ресурсов, вложенные манифесты.

<sup>1</sup> Норенков Ю. И., Усков В. Л. Консультационно-обучающие системы // Вестник МГТУ им. Н.Э. Баумана. [Серия Приборостроение.] 1993. Вып. 3.



Рис. 1. Описание и упаковка учебных материалов

Метаданные учебного курса описываются в соответствии со спецификацией IEEE 1484.12.1–2002 Standard for Learning Object Metadata<sup>2</sup>. Чаще всего это название и описание курса, ключевые слова, направление обучения, уровень сложности и некая информация об авторе.

Информация об организации учебных материалов позволяет на основе одних и тех же ресурсов создать различные способы проведения учебного курса. Например, краткий курс лекций, полный курс лекций, расширенный курс лекций. Имея неупорядоченный набор ресурсов, можно выстраивать различные программы прохождения курса на их основе.

<sup>2</sup> Норенков И.П. Концепция модульного учебника // Информационные технологии. 1996, № 2.

В описании ресурсов содержится список используемых ресурсов. Каждый ресурс может включать произвольное количество файлов с обучающими материалами. Иными словами, ресурс представляет собой ссылки на файлы, которые составляют содержимое курса. Один из этих файлов помечен как основной, т.е. тот, с которого следует начинать отображать данный ресурс. Например, если ресурс выполнен в виде HTML-страницы, он будет содержать HTML-файл в качестве основного, а также все остальные файлы с изображениями, скриптами и т.д., которые необходимы для корректного отображения данной HTML-страницы.

С помощью вложенных манифестов можно объявлять внутри одного манифеста ссылки на другой, что позволяет обобщать учебные материалы и выполнять агрегацию для создания более обобщённых обучающих понятий.

Таким образом, формируется определённый набор файлов, входящих в пакет: все файлы содержимого и файл манифеста. Файлы, представляющие содержимое, могут находиться в любых папках различного уровня вложенности, а файл манифеста *imsmanifest.xml* должен располагаться непосредственно в корневом каталоге курса. После этого все файлы упаковываются в единый PIF-файл (Package Interchange File). В качестве основного формата для PIF-файла рекомендуется использовать архив PKZip (.zip).

Получившийся файл курса легко распространять и использовать, поскольку всё содержимое курса представлено в виде одного файла. Использование же алгоритмов сжатия (в архиваторе ZIP) позволяет экономить дисковое пространство, необходимое для размещения курса.

## Представление содержимого курса

Содержимое учебных материалов может представляться в виде самых разнообразных форматов файлов. Если рассматривать только информационное содержимое, существует несколько подходов к его представлению: отделение содержания от представления; единое представление.

Согласно первому, информационная часть курса должна описываться в строго структурированном виде, обезличенном относительно внешнего вида, т.е. представлять

собой только размеченную информацию (текст, таблицы, формулы, определения) без указания способов её отображения (начертания и размеры шрифтов, внешний вид таблиц и т.д.). Чаще всего в рамках данного подхода предлагается создавать страницы курса в XML, а для их отображения использовать XSL-преобразования и каскадные таблицы стилей (CSS). Это позволяет отображать один и тот же материал разными способами, в зависимости, например, от корпоративного дизайна компании, предоставляющей обучающие сервисы (рис. 2.)

```

...
<раздел имя = «раздел—содержание— разметка»>
  <заголовок> Содержательная (логическая) разметка документа </заголовок>
  Логическая разметка базируется на использовании языков разметки (XML)
  <бибссылка реф = «xml-doc»/>.
</раздел>
<раздел>
  <заголовок> Сервисы и конверторы </заголовок>
  Логическое структурирование документа позволяет использовать множество
  сервисов, например, «умный» поиск и автоматическую каталогизацию. Сервисы можно
  написать на Java <бибссылка реф = "xml-n-java"/>.
</раздел>
<библиография>
  <биб имя = "xml-doc"> Документация по XML. http://www.w3.org </биб>
  <биб имя = "xml-n-java"> М. Дакота, А. Саганич. XML и Java 2. Питер, 2001, 377 с. </биб>
</библиография>

```

### **Раздел 1. Содержательная (логическая) разметка документа**

Логическая разметка базируется на использовании языков разметки (XML)<sup>1</sup>.

### **Раздел 2. Сервисы и конверторы**

Логическое структурирование документа позволяет использовать множество сервисов, например, «умный» поиск и автоматическую каталогизацию. Сервисы можно написать на Java<sup>2</sup>.

Рис. 2. Пример отделения содержания от представления

<sup>1</sup> Документация по XML <http://www.w3.org>

<sup>2</sup> М. Дакота, А. Саганич. XML и Java 2. Издательский дом «Питер», 2001, 377 с.

Основные преимущества такого подхода: возможность автоматического построения содержания, всевозможных индексов и предметных указателей, гибкость настройки стилей отображения. Однако есть и серьёзные недостатки. Во-первых, для подготовки исходного текста необходимо понимать язык разметки XML и иметь под рукой XML-редактор, который позволяет форматировать текст нужными тегами. Во-вторых, конечное отображение материала возможно только при наличии специальных XSLT и CSS файлов, выполняющих необходимые преобразования. В-третьих, набор тегов в исходном XML файле и XSLT таблицах должен быть одним и тем же, а значит, фиксированным. Для редактирования и просмотра таких материалов необходимо пользоваться специальными редакторами, имеющими ограниченный набор тегов. Очевидно, что нельзя сразу предусмотреть весь спектр используемых вариантов разметки, а это означает, что такой подход не будет обеспечивать необходимую функциональность в полном объёме.

Другой же подход, основанный на едином представлении, в данной ситуации оказывается более выгодным. В его основе лежит идея, что учебные материалы при просмотре должны выглядеть в точности так же, как задумывал автор при разработке, т.е. можно использовать любые удобные для автора текстовые редакторы, пользуясь их функциональностью в полной мере. В таком случае отпадает необходимость использовать свои внутренние форматы, а используются широко распространённые по всему миру форматы текстовых файлов, которые могут взаимодействовать с гораздо большим количеством программного обеспечения.

Однако и здесь возникает вопрос: какой же именно формат использовать в качестве основного для хранения информационного содержимого учебных материалов. Наиболее распространённые форматы для представления текстового и графического содержания — HTML, DOC и PDF.

При размещении информации в формате HTML, во-первых, все изображения приходится хранить в отдельных файлах. Во-вторых, результат отображения зависит от настроек браузера, а при изменении размеров клиентского окна браузера форматирование текста также «поплывёт» и может привести к нечитабельному результату. В-третьих, в HTML отсутствует разбивка на страницы и многие элементы оформления (например, колонтитулы). И, наконец, при чтении HTML-страниц, на клиентской машине должны быть установлены все необходимые шрифты, иначе при просмотре неизвестные шрифты будут заменены на стандартные.

Использование формата DOC (Microsoft Word) позволяет решить часть проблем, но имеет свои недостатки. Формат прекрасно хранит как изображения, так и встроенные объекты. Полученный документ имеет строго фиксированный размер страницы и полей. Последние версии редактора Microsoft Word также позволяют выполнять внедрение шрифтов в тело документа. Но поскольку этот формат предназначен в основном для редактирования текста, при просмотре такого файла через Internet Explorer отображается многофункциональная панель форматирования, позволяя пользователю редактировать просматриваемый текст с использованием инструментов Microsoft Word.

Использование же формата PDF (Adobe Acrobat Reader) выгодно отличается от двух предыдущих. Этот формат (Portable Document Format) изначально направлен на идентичное отображение документа на разных платформах. Формат сохраняет любое полиграфическое оформление документа, позволяет исполь-

зовать изображения, обеспечивает хранение содержимого в сжатом виде.

### Выбранная структура учебного курса

В результате была выбрана следующая структура представления учебных материалов (рис. 3).

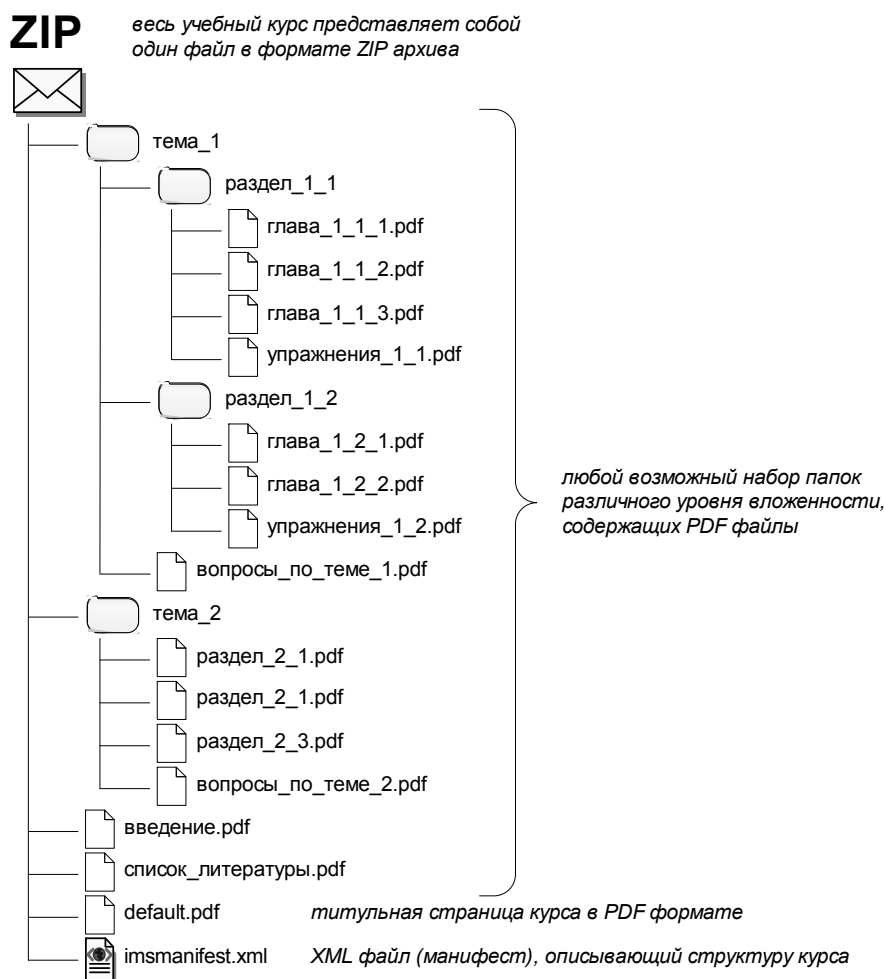


Рис. 3. Структура представления учебного курса

Сам учебный курс представляет собой один файл в формате ZIP-архива. Непосредственно внутри этого файла содержится манифест *imsmanifest.xml* и PDF-файлы, описывающие разделы курса. Последние могут находиться в любых папках любого уровня вложенности. Для удобства разработки и сопровождения такого учебного материала, очевидно, имеет смысл соблюдать иерархическую структуру разделов. Файл *default.pdf* содержит титульную страницу.

В итоге представление учебных материалов в таком формате:

- сохраняет внешний вид материала точно в таком же виде, как было задумано автором;

- позволяет компактно хранить материалы, содержащие разнообразную текстовую и графическую информацию (рисунки, формулы, диаграммы и т.д.);

- позволяет использовать для создания обучающих материалов любые удобные для автора средства (например, Microsoft Word) и их возможности (например, вставку OLE объектов в документ);

- обеспечивает возможность быстро получать готовые материалы из имеющихся документов, хранящихся в другом формате;

- обеспечивает компактное хранение всего курса в одном файле для удобства размещения и использования.

#### **Общая архитектура системы**

Основная часть системы — сервер дистанционного обучения, на котором размещается Web-приложение. Сервер имеет базу данных, которая хранит информацию о пользователях сервера, доступных учебных материалах, накапливает статистическую информацию и содержит ленту новостей.



### Общая архитектура системы

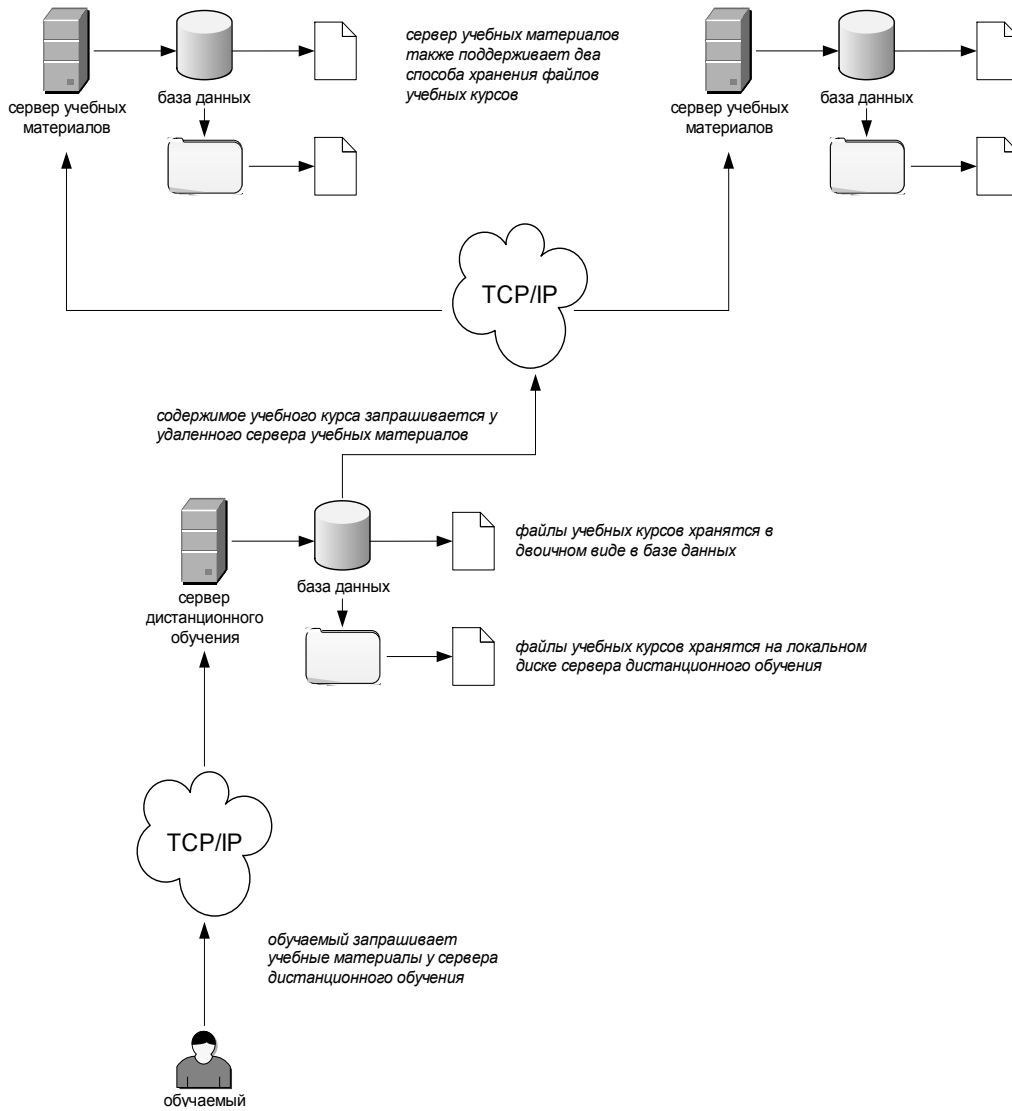


Рис. 4. Общая архитектура системы

## Размещение учебных материалов

С учётом того, что каждый учебный курс представляет собой ZIP-файл, мы получаем следующие преимущества:

— Файлы-контента могут храниться непосредственно на сервере дистанционного обучения. При этом они могут либо находиться на локальном диске (в виде файла), либо быть загружены в базу данных (в двоичном виде). Последняя возможность позволяет не зависеть от файлов на локальном диске, т.е. некоторым образом повышает устойчивость системы.

— Файлы учебных курсов могут находиться на удалённой машине. Для этого используется сервер учебных курсов, который представляет собой Web-сервис. Web-сервис имеет свою собственную базу данных, которая содержит информацию о курсах, размещаемых на этом сервере. Также поддерживается как хранение курсов на локальном диске, так и в двоичном виде в базе данных.

Для отображения содержимого учебных курсов используется специальный модуль, который позволяет отображать только интересующие обучающегося разделы, не загружая весь курс целиком (рис. 5).

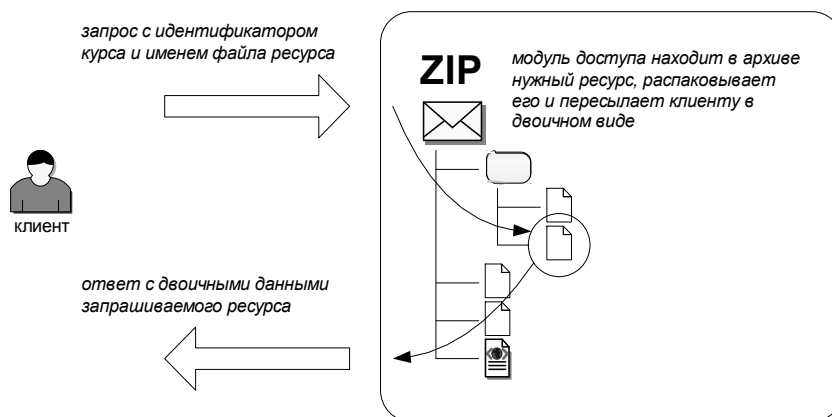


Рис 5. Модуль доступа к учебным материалам

Сервер учебных материалов представляет собой реализацию данного модуля в виде Web-сервиса.

### Разграничение прав доступа

Реализованное Web-приложение имеет средства аутентификации и авторизации для разграничения доступа по следующим ролям (рис. 6):

- гость;
- администратор;
- пользователь.

**Гость** — неавторизованный пользователь. Он может только просматривать новости сайта или войти на сайт в качестве пользователя или администратора.

**Администратор** имеет инструменты для управления учётными записями пользова-

телей, размещения на сайте учебных материалов и назначения пользователей на конкретные курсы. Также администратор может редактировать ленту новостей сайта.

**Пользователь** может просматривать и изучать доступные для него учебные матери-

алы. Также при необходимости предусмотрено изменение некоторых реквизитов своей учётной записи — смена пароля.

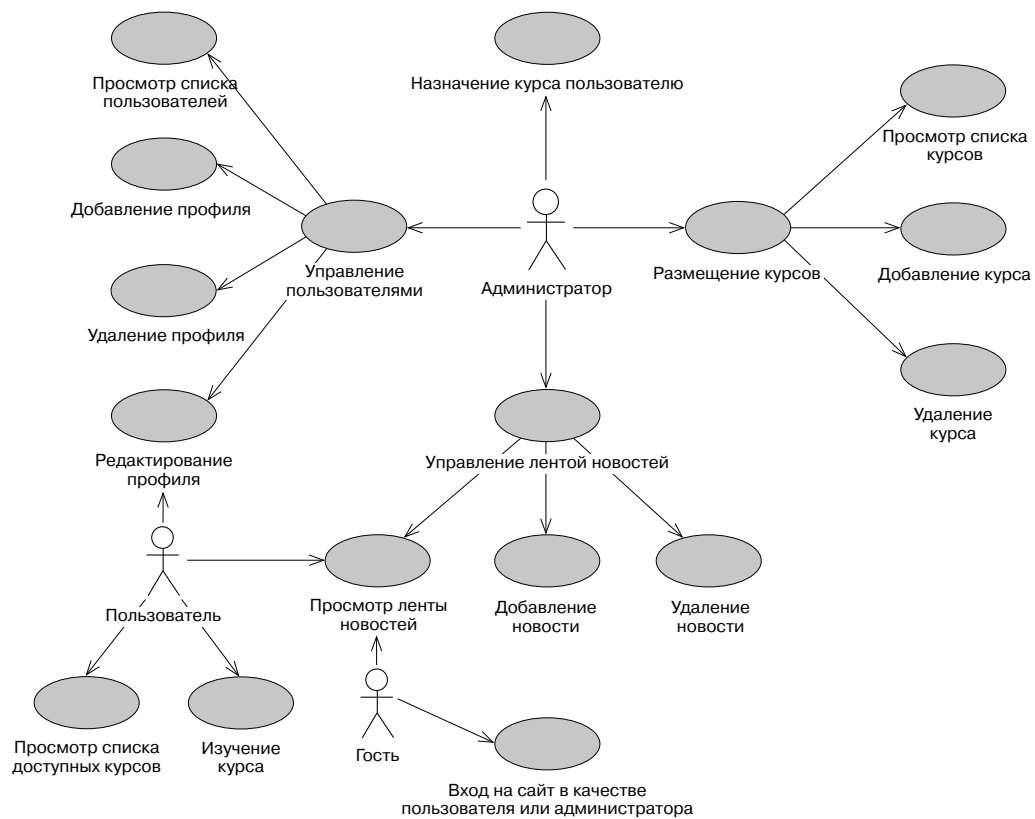


Рис. 6. Диаграмма вариантов использования системы

**Схема базы данных**

Сервер дистанционного обучения имеет базу данных, которая хранит информацию о пользователях сервера, доступных учебных

материалах, накапливает статистическую информацию и содержит ленту новостей. Схема базы данных сервера дистанционного обучения приведена на рис 7.

Схема базы данных сервера дистанционного обучения

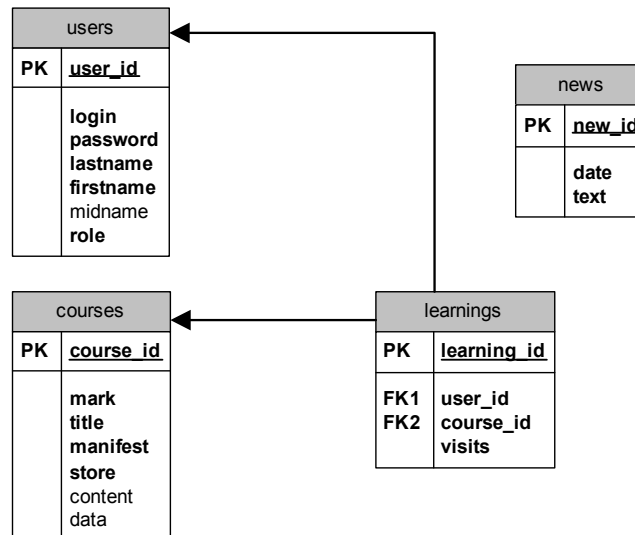


Рис. 7. Схема базы данных сервера дистанционного обучения

Таблица **users** хранит список всех зарегистрированных пользователей. Для каждого пользователя хранятся логин, пароль, фамилия, имя, отчество и роль (пользователь или администратор).

Таблица **courses** хранит список учебных курсов. Для каждого курса хранятся обозначение, наименование, текст манифеста и способ хранения.

**Возможные способы хранения учебных материалов**

**Хранение на сервере дистанционного обучения в базе данных**

При этом в поле **store** заносится код «1», а поле **content** содержит двоичный массив ZIP-файла учебного курса. Поле **data** не используется.

**Хранение на локальном диске сервера дистанционного обучения**

При этом в поле **store** заносится код «2», а поле **data** содержит относительный путь ZIP-файла учебного курса (например, «1\_курс/Введение\_в\_специальность.zip»). Поле **content** не используется.

**Хранение на удалённом сервере учебных курсов**

При этом в поле **store** заносится код «3», а поле **data** содержит URL используемого Web-сервиса и идентификатор курса через разделитель «?» (например, «http://www.mgimt.ru/courses/manager.aspx?5003C266-2BF5-4454-8609-1344E79AC48B»). Поле **content** не используется.

Таблица **learnings** устанавливает соответствие между курсами и назначенными на

них пользователями, а также используется для подсчёта статистики.

Таблица **news** хранит ленту новостей сайта. Для каждой новости хранятся текст новости и дата её поступления.

### Схема базы данных сервера учебных курсов

В основе работы сервера учебных курсов лежит web-сервис, который предоставляет

учебные материалы по запросу. Таким образом, база данных сервера учебных курсов содержит информацию только по хранящимся учебным курсам (рис. 8).

Назначение полей таблицы **courses** полностью аналогично назначению полей соответствующей таблицы в базе данных сервера дистанционного обучения, за исключением того, что доступны два первых способа хранения учебных материалов.

### Физическая структура приложения

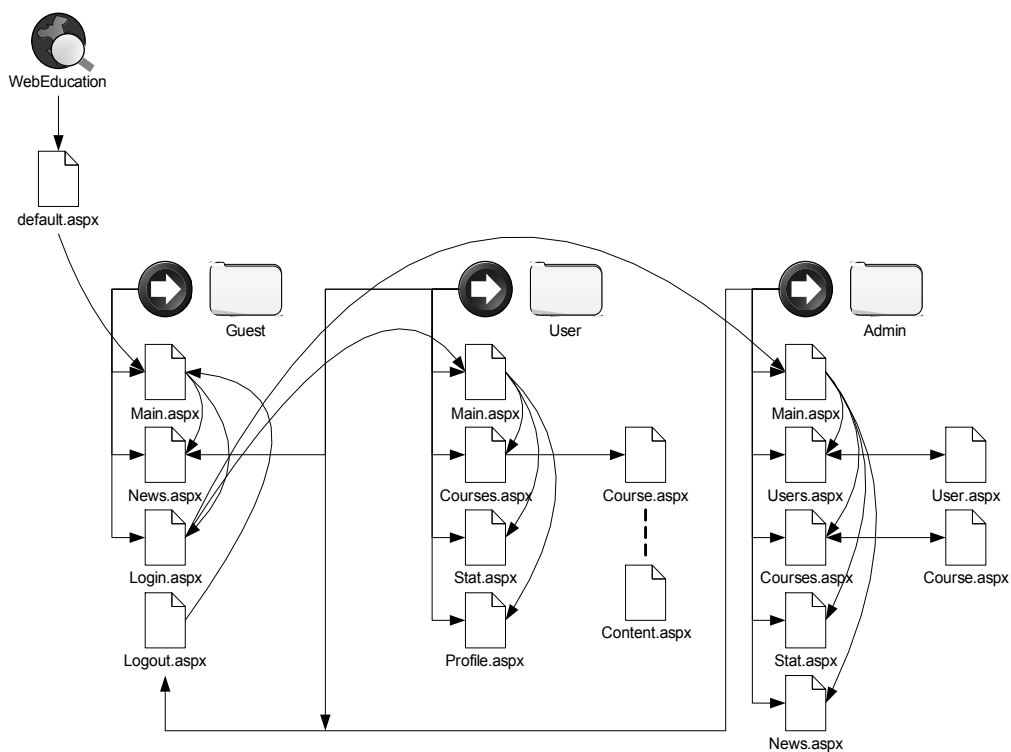


Рис. 8

Рис. 9. Физическая структура размещения ASP.NET страниц



На рис. 9 показана физическая структура размещения ASP.NET страниц в каталогах сервера. Стрелками обозначены возможные переходы с одной страницы на другую. Стрелка в синем кружке обозначает функциональность, доступную с помощью навигационного меню в верхней части страницы, которое имеет различный состав команд в зависимости от роли пользователя.

#### Обозначения и сокращения

API Application Program Interface — программный интерфейс приложений.

CAM — Content Aggregation Model — модель агрегирования учебных материалов (модулей).


LMS — Learning Management System — система управления в обучающих системах.


RTE — Run-Time Environment — среда времени выполнения для обучающих систем.


SCORM — Sharable Content Object Reference Model — объектная модель модульного обучения.

SN Sequencing and Navigation — модель упорядочивания и управления.

XML Extensible Markup Language — расширяемый язык разметки.

 **Норенков Ю.И., Усков В.Л.** Консультационно-обучающие системы // Вестник МГТУ им. Н.Э. Баумана. [Серия Приборостроение]. 1993. Вып. 3.

 **Норенков И.П.** Концепция модульного учебника // Информационные технологии. 1996, № 2.

 **Рогозин О.В.** Основные принципы организации моделей SCORM в системах электронного обучения. Новые информационные технологии: Материалы восьмого научно-практического семинара. М.: Моск. гос. ин-т электроники и математики, 2005.