# РЕФЕРАТ

**Записка:** 83 стр., 30 рис., 4 табл., 1 приложение, 28 источников.

Объект исследования – информационная система социальной сети.

**Цель работы** — Разработка и программная реализация информационной систепмы социальной сети спортивного сайта с использованием современных технологий

Методы исследования — теория множеств, теории реляционных баз данных, модель "сущность-связь".

Результаты — проведен обзор существующих решений для данной задачи, в результате которого подходящей системой управления контента была выбрана Joomla. Проведен обзор современных социальных веб сервисов для социальной сети и были выбраны популярные и наиболее подходящие для работы нишевой социальной сети. На следующем этапе была составлена и проработана информационная модель социальной сети. Так же была разработана и нормализованна база данных и конмплекс модулей и компонентов для информационной системы социальной сети спортивной тематики.

ИНФОРМАЦИОННАЯ СИСТЕМА СОЦИАЛЬНОЙ СЕТИ,

НИШЕВАЯ СОЦИАЛЬНАЯ СЕТЬ, СИСТЕМА УПРАВЛЕНИЯ КОНТЕНТОМ, mvc, ajax, php, МЕТОД SLIDING DOORS, XML, CSS, Joomla,jcomment, jQUERY FANCYBOX, IntelliTXT.

**Содержание**

[ВВЕДЕНИЕ 4](#_Toc230574003)

[1 ИНФОРМАЦИОННЫЙ ОБЗОР 6](#_Toc230574004)

[1.1 Классификация социальных сетей 6](#_Toc230574005)

[1.2 Системы управления контентом и их особенности 10](#_Toc230574006)

[1.3 Разработка информационных ресурсов с использованием технологии Ajax 12](#_Toc230574007)

[1.4 Разработка информационных ресурсов с использованием технологии ХМL 18](#_Toc230574008)

[1.5 Постановка задачи 21](#_Toc230574009)

[2 ВЫБОР МЕТОДА РЕШЕНИЯ ПОСТАВЛЕННОЙ ЗАДАЧИ 24](#_Toc230574010)

[2.1 Использование языка PHP для разработки веб приложений 24](#_Toc230574011)

[2.2 Выбор системы управления контентом 27](#_Toc230574012)

2.3 Анализ и выбор основных социальных сервисов для социальной сети………………………………………………………...……….31

[3 ПРОГРАММНАЯ РЕАЛИЗАЦИЯ 34](#_Toc230574013)

[3.1](#_Toc230574014) Информационная модель приложения…………………………...34

[3.2](#_Toc230574015) Программная реализация компонента «Home» 36

[3.3](#_Toc230574016) Программная реализация компонента «Profile» 44

[3.4](#_Toc230574017) Программная реализация компонента «Friends» 56

[3.5 Программная реализация компонента «Applications»………...…58](#_Toc230574018)

[3.6 Программная реализация компонента «Mail» 63](#_Toc230574018)

[4 ОХРАНА ТРУДА 66](#_Toc230574019)

[4.1 Характеристика помещения 66](#_Toc230574020)

[4.2 Анализ состояния охраны труда 69](#_Toc230574021)

[4.3 Рекомендации по улучшению рабочего помещения 79](#_Toc230574022)

[ВЫВОДЫ 81](#_Toc230574023)

[СПИСОК ЛИТЕРАТУРЫ 82](#_Toc230574024)

[ПРИЛОЖЕНИЕ 84](#_Toc230574025)

**ВВЕДЕНИЕ**

Активность пользователей Интернета возрастает с устойчивым постоянством. Этому способствует в первую очередь доступность современных технологий, но не в меньшей степени и развитие самого виртуального пространства: различные онлайн-сервисы, бесплатные развлекательные ресурсы, социальные сети. Наиболее важная составляющая развития телекоммуникационных сетей это возможность непосредственного общения людей. Организация общения в интернете происходит с помощью основных 3-х современных веб технологий: гостевые книги, форумы, блоги. С развитием этих форм общения стали социальные сети —сообщесва участников, объединенных не только средой общения, но и с явно установленными связями между собой.

Социальные сети сегодня уже посещает более чем две трети онлайновой аудитории во всем мире, и это четвертая по популярности онлайновая категория после поиска, порталов и программного обеспечения, которая опережает электронную почту.

Каждый день в Интернете появляется сразу несколько тематических, нишевых социальных сетей, которые становятся местом, где встречаются и проводят время все больше людей. Это не просто пассивные зрители, они непосредственные участники, и даже больше – создатели своих личных, вполне реальных миров.

Социальные сети служат площадкой для неформального общения приятелей подобно LiveJournal, помогают создавать новую музыку как MySpace, расширяют игровой опыт массовых онлайн-игр в сети Xfire. А так же служат серьезным инструментом для поиска сотрудников и партнеров в LinkedIn. Таким образом, социальные сети являются ключевым элементом для дальнейшего развития сообществ и в скором времени они будут неотъемлемым инструментом для любой деятельности.

В данной работе рассматриваеться разработка и программная реализация нишевой социальной сети спортивной тематики. Для этого проводится анализ наиболее популярных систем представителей социальных сетей, который раскрывает ряд недостатков их систем управления контентом и позволяет разработать и программно реализовать дополнительный комплекс модулей для подобных систем.

# 1 ИНФОРМАЦИОННЫЙ ОБЗОР

## 1.1 Классификация социальных сетей

Сервис социальной сети – это программный комплекс, платформа для взаимодействия людей в группе или группах. Заданием такого сервиса является обеспечение пользователей широким спектром средств коммуникации, как то видео, изображение, музыка, блоги, форумы и т.д.

Первая социальная сеть была создана в 1995 году. Новому направлению веб-сервисов дал начало американский сайт Classmates.com, который предлагал найти своих одноклассников, коллег и друзей. Сейчас в сети Classmates.com зарегистрировано более 40 миллионов активных пользователей из США и Канады.

XXI век стал расцветом подобных сервисов. 2001 году появилась первая социальная сеть, направленная на профессиональное развитие – Ryze (~500 тысяч пользователей); 2003 год – появление LinkedIn (~20 миллионов пользователей); 2004 год – orkut (~16 миллионов пользователей), Facebook (~73 миллионов пользователей), Friendster (~50 миллионов), MySpace и другие. Головокружительный успех MySpace в 2005 году (за два года существования сервиса его аудитория выросла до 47,3 миллионов людей) привел к быстрому разрастанию социальных сетей и увеличению их популярности среди пользователей.

Всесоциальные сети можно условно разделить на три категории[6]:

1. Общей тематики (созданные для приватного общения)
2. Специализированные
3. Деловые

Используя приведенную классификацию представим наиболее известные социальные сервисы на рисунке рисунке 1.1.

Рисунок 1.1 — Схема классификации социальных сервисов.

Социальные сети **общей тематики**[6], например, Facebook, Вконтакте и Одноклассники, направлены на поддержание и восстановление уже существующих контактов и поиск новых знакомых. Основной аудиторией являются школьники и студенты, которые используют веб-сервисы этого типа как платформу для общения в сети. Эти сервисы уже накопили огромную базу контактов и профилей, из которых можно многое узнать о человеке и его предпочтениях. Большая пользовательская база и эффект «вирусного маркетинга» позволяет сохранять высокие темпы прироста пользователей.

**Специализированные** социальные сети занимают определенную нишу и собирают аудиторию по интересам. Например, Last.Fm – музыкальная социальная сеть, которая развивает связи между пользователями по принципу музыкальных предпочтений. Geni – семейная социальная сеть, объединяет людей на основе родственных связей. Социальная сеть MyChurch предоставляет возможность любому человеку создать свою церковь и свой круг общения. Autokadabra – социальная сеть для автолюбителей. Habrahabr – сообщество специалистов в сфере компьютерных технологий.

**Деловые социальные сети**, такие как LinkedIn, Plaxo, МойКруг и Вштате, направлены на профессиональное развитие и построение карьеры. Новой в этом списке является деловая социальная сеть Profeo, запуск которой состоялся в апреле 2008 года. У Profeo есть украиноязычный интерфейс. Специфика Profeo состоит в уникальном для Украины сочетании разных услуг и сервисов, доступных на одном ресурсе: онлайн-адресная книжка, блоги, форумы, мгновенные сообщения и группы по профессиональным интересам.

Общая схема взаимодействия пользователя социальной сети и его друзей в ней представленная на рисунке 1.2

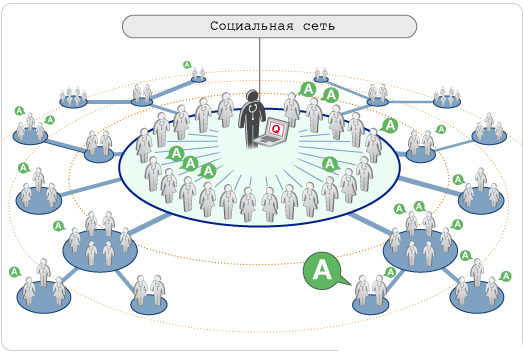


Рисунок. 1.2 - Схема взаимодействия пользователя социальной сети и его друзей в ней

С этой схемы видно что пользователи социальной сети могут быть связаны с другими людьми которые находиться как в других группах, так и других социальных сетях. Как это представлено в социальных сетях Myspace, Twitter.

Крупнейшей социальной сетью на данный момент является, MySpace. Аудитория этого ресурса составляют более 130 миллионов человек. Благодаря процессам информатизации и более тесного интеграции человека в информационную среду социальные сети быстро набирают популярность. Эта тенденция наблюдается и в Украине. Последние исследования компании InMind за IV квартал 2007 года показывают, что 26% населения Украины в возрасте от 15 лет (то есть 7 миллионов людей) являются активными пользователями Всемирной сети (пользуются Интернетом чаще 1 раза в месяц). По данным компании GfK, эта цифра составляет 6,3 миллиона пользователей или 16% всего населения Украины. Быстрое проникновение Интернета способствует географическому расширению сети личных контактов. Люди становятся более мобильными – легко меняют работу, место проживания, но стремятся сохранить отношения с друзьями, коллегами, единомышленниками. Поэтому в мире, где на общение остается все меньше времени, социальные сети кажутся одним из наиболее удобных инструментов для нахождения бывших коллег, одноклассников и людей с общими интересами.

## 1.2 Системы управления контентом и их особенности

Система управления контентом (Content Management System, CMS) - пакет программ, специально разработанный для автоматизации процессов управления информационными ресурсами в сети интернет. Основная их задача – обеспечить необходимые возможности по работе с определенным типом контента (который зависит от специфики сайта) и абстрагировать от пользователя (и администратора) нюансы используемой аппаратуры, сервера, языка программирования и т.д.

Использование CMS предоставляет возможность простого обновления контента без необходимости в специальных знаниях в области веб-технологий. Зачастую для этого используется веб-интерфейс реализующий возможности WYSIWYG способ редактирования, при котором редактируемый материал в процессе редактирования выглядит в точности так же, как и конечный результат. Использование подобных продуктов позволяет легко добавлять новые страницы и удалять старые или даже изменять структуру сайта в соответствии с изменениями в направлениях ведения бизнеса. CMS также автоматизирует выполнение разных заданий, связанных с оформлением сайта, таких как единое оформление на всех страницах сайта, меню и система навигации[2].

Основной целью создания CMS – сконцентрировать действия пользователя на создании контента, а не технологиях его последующего размещения. Системы управления контентом очень упрощают процесс работы с веб-приложениями. Не нужно знать множество процедур и алгоритмов. Любой человек может вносить изменения на сайте без знаний HTML разметки. Изменения на сайте могут быть осуществлены в любое время. Это очень важно, в том случае, если сайт является важной частью маркетинговой политики организации и каналом связи с клиентом.

Использование CMS позволяет распределить работу по наполнению сайта между группой людей с учетам прав их доступа к отдельным элементам. Вместо того что бы один человек занимался развитием сайта, это могут делать несколько людей одновременно, при этом CMS отследит вклад каждого. Также администратор сайта может быть уверен в том что, каждый сотрудник штата может обновлять только ту секцию, которая ему доступна согласно прав доступа.

В результате анализа систем управления контентам были выявлены требования, предъявляемые к ним с целью достижения наиболее гибкой настройки, стабильности работы и универсальности. Система управления контентом должна:

* иметь модульную структуру;
* регулярное обновление ядра системы и круглосуточную техническую поддержку;
* иметь открытый исходный код. Это дает возможность быстрей находить ошибки и исправлять их не только производителями продукта, но и самими пользователями;
* позволять поэтапного наращивания производительности системы (маштабируемость);
* основываться на современных технологий, таких как PHP, MySQL, XML, AJAX;
* включать систему кэширования, быть устойчивой к нагрузкам.

## 1.3 Разработка информационных ресурсов с использованием технологии Ajax

Когда существующих возможностей становиться мало, а совершенствовать существующее, уже некуда, тогда и происходит технологический прорыв. Таким прорывом и есть AJAX (Asynchronous JavaScript and XML) - подход к построению пользовательских интерфейсов веб-приложений, при котором веб-страница, не перезагружаясь, сама догружает нужные пользователю данные.

Ajax не является самостоятельной технологией. В сущности, это объединение нескольких самостоятельных технологий, которые могут быть эффективно использованы вместе. Ajax включает в себя[14]:

* Стандартные средства отображения страниц, такие как XHTML и CSS
* Динамические средства отображения информации и взаимодействия с пользователем - Document Object Model
* Обмен данными и их обработка - XML и XSLT
* Механизмы асинхронной передачи данных с сервера с помощью XMLHttpRequest
* JavaScript, который объединяет всё это вместе.

В настоящее время разработка веб-приложений стремится к разграничению хранения данных, их обработки и отображения. Проекты становятся сложнее, и переплетать между собой различные технологии становиться слишком дорого для времени разработчика. Так, например, все стили форматирования выносятся в CSS или в XSL файлы, HTML или XML данные хранятся в других разделах, серверные обработчики в третьих, базы данных в четвертых. И если еще 5-6 лет назад практически везде можно было увидеть переплетение всего этого в одном файле, то сейчас это все чаще становиться редкостью.

**Model-view-controller** (**MVC**, «Модель-представление-поведение») — архитектура программного обеспечения, в которой модель данных приложения, пользовательский интерфейс и управляющая логика разделены на три отдельных компонента, так, что модификация одного из компонентов оказывает минимальное воздействие на другие компоненты[5].

Шаблон MVC позволяет разделить данные, представление и обработку действий пользователя на три отдельных компонента

* Модель (Model). Модель предоставляет данные (обычно для View), а также реагирует на запросы (обычно от контроллера), изменяя свое состояние.
* Представление (View). Отвечает за отображение информации (пользовательский интерфейс).
* Поведение (Controller). Интерпретирует данные, введенные пользователем, и информирует модель и представление о необходимости соответствующей реакции.

Взаимодействие перечисленных компонентов представлено на рисунке 1.3.

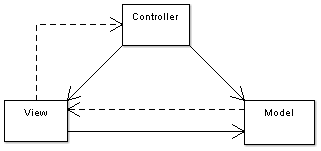


Рисунок 1.3 – Схема взаимодействия компонентов модели MVC

Представление, так и поведение зависят от модели. Однако модель не зависит ни от представления, ни от поведения. Это одно из ключевых достоинств подобного разделения. Оно позволяет строить модель независимо от визуального представления, а также создавать несколько различных представлений для одной модели.

При разработке более сложных проектов возникает необходимость в структурированности и удобно читаемости кода. Не следует засорять код программиста кодом верстальщика, а код верстальщика - правками дизайнера.

Возникает необходимость в разграничении работы. Так, например, дизайнер будет делать свою работу, верстальщик свою, программист свою, и при этом никто друг другу мешать не будет. В итоге каждому участнику проекта достаточно будет знать только те данные, с которыми ему придется работать. В таком случае производительность группы и качество проекта повышается в разы. В настоящее время эта проблема с успехом решается путем использования шаблонов, однако это тоже создает определенные трудности, так как чтобы, скажем, подключить Smarty, необходимо подключить программный модуль обработки шаблонов, и четко связать со структурой проекта. А это далеко не всегда возможно и требует определенных затрат. Немного проще при использовании связки XML + XSL, так как они предоставляют больше возможностей, но и создают дополнительные трудности.

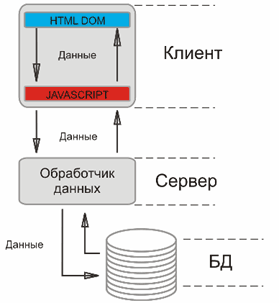
Ajax - это не самостоятельная технология. Это идея, которая базируется на двух основных принципах:

* использование DHTML для динамичного изменения содержания страницы;
* использование XMLHttpRequest для обращения к серверу "на лету".

Объединение, этих двух подходов позволяет создавать намного более удобные веб-интерфейсы пользователя на тех страницах сайтов, где необходимо активное взаимодействие с пользователем. Использование Ajax стало наиболее популярно после того, как компания Google начала активно использовать его при создании своих сайтов, таких как Gmail, Google maps и Google suggest. Создание этих сайтов подтвердило эффективность использования данного подхода.

Рассмотрим как происходит взаимодействие между сервером и клиентом с использованием Ajax. На рисунке 1.4 приведена классическая модель веб-приложения.

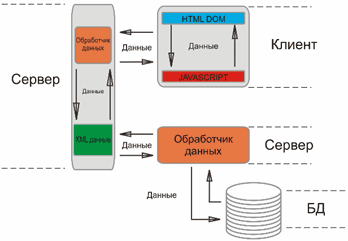
Рисунок 1.4 — Модель веб-приложения



Клиент, набирая в строке поиска адрес интересующего его ресурса, попадая на сервер, делает к нему запрос. Сервер производит вычисления в соответствии с запросом, обращается к базе данных и так далее, после чего полученные данные идут клиенту и, в случае необходимости подставляются в шаблоны и обрабатываются браузером. Результатом является страница, которую мы видим. Это самая простая модель взаимодействия и, как следствие, самая распространенная [16].

Модель взаимодействия с использованием Ajax рассмотрена на рисунке 1.4

Рисунок 1.5— Модель взаимодействия AJAX



Последовательность действий клиента сохраняется и он, скорее всего не поймет того, что будет происходить. Но на стороне сервера все выглядит иначе. При обращении к серверу, генерируется страница, которая будет отображаться пользователю, и предлагать ему совершить интересующую его последовательность действий. При сознательном выборе клиента, его запрос будет обращаться к AJAX модулю, который и будет производить все интересующие его вычисления и работу с сервером как таковым. Основное отличие в том, что этот метод дает нам возможность динамически обращаться к серверу и выполнять интересующие нас действия.

Приведем основные преимущества рассмотренной технологии[17].

* Экономия трафика. Использование AJAX позволяет значительно сократить трафик при работе с веб-приложением благодаря тому, что часто вместо загрузки всей страницы достаточно загрузить только изменившуюся часть, иногда довольно небольшую.
* Уменьшение нагрузки на сервер. AJAX позволяет несколько снизить нагрузку на сервер. К примеру, на странице работы с почтой, когда вы отмечаете прочитанные письма, серверу достаточно внести изменения в базу данных и отправить клиентскому скрипта сообщение об успешном выполнении операции без необходимости повторно создавать страницу и передавать её клиенту.
* Ускорение реакции интерфейса. Поскольку нужно загрузить только изменившуюся часть, то пользователь видит результат своих действий быстрее.

Однако данная технология обладает и рядом недостатков, рассмотренных ниже.

* Интеграция со стандартными инструментами браузера. Динамически создаваемые страницы не регистрируются браузером в истории посещения страниц, поэтому не работает кнопка «Назад», предоставляющая пользователям возможность вернуться к просмотренным ранее страницам, но существуют скрипты, которые могут решить эту проблему. Другой недостаток изменения содержимого страницы при постоянном URL заключается в невозможности сохранения закладки на желаемый материал. Частично решить эти проблемы можно с помощью динамического изменения идентификатора фрагмента (части URL после #), что позволяют многие браузеры.
* Динамически загружаемое содержимое недоступно поисковикам (если не проверять запрос, обычный он или XHttpRequest). Поисковые машины не могут выполнять JavaScript, поэтому разработчики должны позаботиться об альтернативных способах доступа к содержимому сайта.
* Старые методы учёта статистики сайтов становятся неактуальными. Многие сервисы статистики ведут учёт просмотров новых страниц сайта. Для сайтов, страницы которых широко используют AJAX, такая статистика теряет актуальность.
* Требуется обязательная поддержка JavaScript. Если в браузере отсутствует поддержка JavaScript, то это делает невозможной работу с AJAX. Возможные проблемы при работе на браузерах старых версий. Проблемы с производительностью, увеличение количества JavaScript кода на странице замедляет работу браузера, хотя при умном планировании и современных компьютерах такая проблема маловероятна.

## 1.4 Разработка информационных ресурсов с использованием технологии ХМL

**XML** (англ. *eXtensible Markup Language* — расширяемый язык разметки) — рекомендованный Консорциумом Всемирной паутины язык разметки, фактически представляющий собой свод общих синтаксических правил. XML — текстовый формат, предназначенный для хранения структурированных данных (взамен существующих файлов баз данных), для обмена информацией между программами, а также для создания на его основе более специализированных языков разметки (например, XHTML), иногда называемых *словарями*[8]. XML является упрощённым подмножеством языка SGML.

Целью создания XML было обеспечение совместимости при передаче структурированных данных между разными системами обработки информации, особенно при передаче таких данных через Интернет. Словари, основанные на XML (например, RDF, RSS, MathML, XHTML, SVG), сами по себе формально описаны, что позволяет программно изменять и проверять документы на основе этих словарей, не зная их семантики, то есть не зная смыслового значения элементов. Важной особенностью XML также является применение так называемых пространств имён.

В отличие от HTML, XML сам по себе не содержит никаких тэгов, предназначенных для разметки, он просто определяет порядок их создания. XML-тэги предназначены для определения элементов документа, их атрибутов и других конструкций языка. Таким образом, у разработчиков появляется уникальная возможность определять собственные команды, позволяющие им наиболее эффективно определять данные, содержащиеся в документе. Автор документа создает его структуру, используя те команды, которые удовлетворяют его требованиям, и добивается такого типа разметки, которое необходимо ему для выполнения операций просмотра, анализа документа и поиска в нем.

Перечислим достоинства и недостатки XML. К достоинствам этого языка разметки можно отнести следующее:

* XML — язык разметки, позволяющий отобразить двоичные данные в текст, читаемый человеком и анализируемый компьютером;
* XML поддерживает Юникод;
* в формате XML могут быть описаны такие структуры данных как записи, списки и деревья;
* XML — это самодокументируемый формат, который описывает структуру и имена полей также как и значения полей;
* XML имеет строго определённый синтаксис и требования к анализу, что позволяет ему оставаться простым, эффективным и непротиворечивым. Одновременно с этим, разные разработчики не ограничены в выборе экспрессивных методов (Например, можно моделировать данные помещая значения в параметры тегов или в тело тегов, можно использовать различные языки и нотации для именования тэгов);
* XML — формат, основанный на международных стандартах;
* иерархическая структура XML подходит для описания практически любых типов документов, кроме аудио и видео мультимедийных потоков, растровых изображений, сетевых структур данных и двоичных данных;
* XML представляет собой простой текст, свободный от лицензирования и каких-либо ограничений;
* XML не зависит от платформы;
* XML является подмножеством SGML (который используется с 1986 года). Уже накоплен большой опыт работы с языком и созданы специализированные приложения;
* XML не накладывает требований на расположение символов в строке;
* в отличие от бинарных форматов, XML содержит метаданные о именах, типах и классах описываемых объектов, по которым приложение может обработать документ неизвестной структуры (например, для динамического построения интерфейсов);
* XML поддерживается на низком аппаратном, микропрограммном и программном уровнях в современных аппаратных решениях.

Недостатками данного представления данных является следующее.

* Синтаксис XML избыточен.
* Размер XML документа существенно больше бинарного представления тех же данных. В грубых оценках величину этого фактора принимают за 1 порядок (в 10 раз).
* Размер XML документа существенно больше, чем документа в альтернативных текстовых форматах передачи данных (например JSON) и особенно в форматах данных, оптимизированных для конкретного случая использования.
* Избыточность XML может повлиять на эффективность приложения. Возрастает стоимость хранения, обработки и передачи данных.
* XML содержит метаданные (об именах полей, классов, вложенности структур), и одновременно XML позиционируется как язык взаимодействия открытых систем. При передаче между системами большого количества объектов одного типа (одной структуры), передавать метаданные повторно нет смысла, хотя они содержатся в каждом экземпляре XML описания.
* Для большого количества задач не нужна вся мощь синтаксиса XML и можно использовать значительно более простые и производительные решения.
* XML не содержит встроенной в язык поддержки типов данных. В нём нет строгой типизации, то есть понятий «целых чисел», «строк», «дат», «булевых значений» и т.д.
* Иерархическая модель данных, предлагаемая XML, ограничена по сравнению с реляционной моделью и объектно-ориентированными графами и сетевой моделью данных.

В XML предусмотрено использование специальных инструкций, управляющих процессом отображения элементов в окне программы-клиента (в окне браузера). Для этих целей в HTML, используются CSS (Cascade Style Sheets), а в XML - стилевые таблицы XSL (eXtensible Stylesheet Language).

**XSL** (Extensible Stylesheet Language) — расширяемый язык таблиц стилей. Основные отличия от CSS.

* CSS и XSL — принципиально разные технологии. Их области применения пересекаются лишь частично.
* CSS-форматирование применяется к HTML-документу браузером на клиентской стороне, а XSL-преобразование выполняется, как правило, на сервере, затем его результат отправляется в браузер клиента.
* Языком, лежащим в основе XSL, является XML, а это означает, что XSL более гибок, универсален, и у разработчиков появляется возможность использования средств контроля за корректностью составления таких стилевых списков (используя DTD или схемы данных).