Содержание:

Введение

Глобальная сеть - совокупность компьютеров, расположенных на больших расстояниях друг от друга, а также система каналов передачи связи: средств коммуникации (переключения), обеспечивающих соединение пользовательских коммуникационных систем и обмен данными между ними.[1]

Глобальные сети (WideAreaNetworks, WAN) создаются крупными телекоммуникационными компаниями для оказания платных услуг абонентам.

Интернет - мировая глобальная компьютерная сеть. Она составлена из разнообразных компьютерных сетей, объединенных стандартными соглашениями о способах обмена информацией и единой системой адресации. Интернет использует протоколы семейства TCP/IP. Они хороши тем, что обеспечивают относительно дешевую возможность надежно и быстро передавать информацию даже по не слишком надежным линиям связи, а также строить программное обеспечение, пригодное для работы на любой аппаратуре. Система адресации (URL-адреса) обеспечивает уникальными координатами каждый компьютер (точнее, практически каждый ресурс компьютера) и каждого пользователя Интернета, создавая возможность взять именно то, что нужно, и передать именно туда, куда нужно.

В 1994 году началась революция – World Wide Web. Всемирная паутина World Wide Web (WWW) соткана из Web-страниц, которые содержат в себе разную информацию в зависимости от тематики Web сайта. В основу Web был положен гипертекст (hypertext) – метод связывания блоков, или «страниц», данных, придуманный еще в шестидесятых годах. Однако только в девяностых годах Бернерс-Ли и его сотрудники перенесли концепцию гипертекста в Internet, создав HTTP – Hypertext Transfer Protocol (протокол передачи гипертекста). С появлением HTTP родился и World Wide Web.

Сегодня в Internet существуют миллионы Web – сайтов. Можно получать доступ к информации по различным темам, открыть в Web свой бизнес. Более того, в Web можно найти и сведения о нем самом и о тех технологиях, на которых он основан.

Таким образом, данная курсовая работа носит актуальный характер. Целью работы является обзор языков гипертекстовой разметки. Для достижения данной цели требуется решить следующие задачи:

1. провести анализ понятия «гипертекст»;
2. проанализировать историю развития гипертекста;
3. провести анализ моделей гипертекста;
4. изучить виды языков гипертекстовой разметки документов – синтаксис, структуру документов, основные элементы.
5. провести анализ будущего Web-программирования.

Глава 1. Гипертекст

1.1 Понятие гипертекста

Гипертекст - текст со вставленными в него словами (командами) разметки, ссылающимися на другие места этого текста, другие документы, картинки и т.д. Во время чтения такого текста (в соответствующей программе, его обрабатывающей и выполняющей соответствующие ссылки или действия) вы видите подсвеченные (выделенные) в тексте слова. [3] Если наехать на них курсором и нажать клавишу или на кнопку (глаз) мышки, то высветится то, на что ссылалось это слово, например, другой параграф той же главы этого же текста. [1] В WWW по ключевым словам можно попасть в совершенно другой текст из другого документа, войти в какую-нибудь программу, произвести какое-либо действие и т.д. В Internet в контексте WWW можно получать доступ к чему угодно, к telnet, e-mail, ftp, Gopher, WAIS, Archie, USENET News и т.п. В WWW можно ссылаться на данные на других машинах в любом месте сети, тогда при активации этой ссылки эти данные автоматически передадутся на исходную машину и вы увидите на экране текст, данные, картинку, а если провести в жизнь идею мультимедиа, то и звук услышите, музыку, речь. [2] Это слегка напоминает Gopher, но фактически это принципиально другое и новое. В Gopher имеется жесткая структура меню, по который вы двигаетесь, как вам угодно. Эта структура не зависит от того, что вы делаете, какой документ пользуете и т.д. В WWW вы двигаетесь по документу, который может иметь какую угодно гипертекстовую структуру. Можно свободно организовать структуры меню в гипертексте. Имея редактор гипертекстов, можно создать любую структуру рабочей среды, включая документацию, файлы, данные,

картины, программное обеспечение и т.д., и это не будет новое программное обеспечение, а просто гипертекст.

Гипертекстовая технология – это представление текста в виде многомерной иерархической структуры типа сети.[1]

Гипертекст формируется в результате представлений текста как ассоциативно связанных блоков информации. Ассоциативная связь – это соединение, сближение представлений, смежных, противоположных, аналогичных. Гипертекст значительно отличается от обычного текста. Обычные (линейные) тексты имеют последовательную структуру и предусматривают их чтение слева направо и сверху вниз. [2]

Простейший пример гипертекста - это любой словарь или энциклопедия, где каждая статья имеет отсылки к другим статьям этого же словаря. В результате читать такой текст можно по-разному: от одной статьи к другой, по мере надобности, игнорируя гипертекстовые отсылки; читать статьи подряд, справляясь с отсылками; наконец, пуститься в гипертекстовое плавание, то есть от одной отсылки переходить к другой. [2]

Концепция гипертекста достаточно проста. Есть база данных, в базе данных находятся объекты. Объекты это, чаще всего, небольшие текстовые разделы, посвященные тому или иному вопросу. Специальные механизмы и правила позволяют компьютеру поддерживать ссылки из одних текстовых фрагментов в другие. Человек или программный агент может устанавливать новые связи между текстовыми фрагментами. Система текстовых фрагментов или файлов с такой организацией получила название "гипертекст". [3]

Гипертекст изначально создавался как среда поддерживающая взаимодействие нескольких людей. Культовая работа Ваннавера Буша "As we may Think", в которой он описал устройство Memex, была связаны с проблемами взаимодействия коллективов ученых после Второй Мировой Войны, когда стало ясно, что существующие системы плохо поддерживают коллективную мыслительную деятельность. Система Memex, по своей сути, представляла систему для обмена "мемами" - элементарными единицами культурной эволюции. Гипертекст изначально мыслился создателям как система общественной деятельности. Группа взаимосвязанных сообщений образовывала сеть, и эта гипертекстовая сеть документов поддерживала социальную сеть отношений между сообществом авторов коллективного гипертекста.[3]

Использование гипертекста позволяет фиксировать отдельные идеи, мысли, факты, а затем связывать их друг с другом, двигаясь в любых направлениях, определяемых ассоциативными связями.

С развитием компьютерных средств мультимедиа гипертекст начал превращаться в более наглядную информационную форму, получившую название гипермедиа — эта информационная форма содержит не только текст, но и графику, видеоинформацию и звуки.

Обработка гипертекста открыла новые возможности освоения информации, качественно отличающиеся от традиционных способов.

Вместо поиска информации по соответствующему поисковому ключу гипертекстовая технология предполагает перемещение от одних объектов информации к другим с учетом их смысловой, семантической связанности.

Обработке информации по правилам формального вывода в гипертекстовой технологии соответствует запоминание пути перемещения по гипертекстовой сети.[2]

Гипертексты обладают определенной семантической (смысловой) сетевой структурой. При многократном просмотре, если гипертекст используется как учебник, эта структура будет сильно влиять на структуру знаний пользователя по изучаемому вопросу. Поэтому при построении гипертекстовых систем следует уделять внимание не только тому, как разбить исходный текст на части, но и тому, насколько пользователю будет понятно, легко и удобно работать с этими частями текста.

Структурно гипертекст состоит из информационного материала, тезауруса гипертекста, списка главных тем и алфавитного словаря.[2]

Информационный материал подразделяется на информационные статьи, состоящие из заголовка статьи и текста. Заголовок содержит тему или наименование описываемого объекта.

Информационная статья содержит традиционные определения и понятия, должна занимать одну панель и быть легко обозримой, чтобы пользователь мог понять, стоит ли ее внимательно читать или перейти к другим, близким по смыслу статьям.

Текст, включаемый в информационную статью, может сопровождаться пояснениями, примерами, документами, объектами реального мира.[2]

Тезаурус гипертекста – это автоматизированный словарь, отображающий семантические отношения между лексическими единицами дескрипторного информационно-поискового языка и предназначенный для поиска слов по их смысловому содержанию.[2]

Тезаурус гипертекста можно представить в виде сети: в узлах находятся текстовые описания объекта (информационные статьи), ребра сети указывают на существование связи между объектами и на тип родства.[2]

Список главных тем содержит заголовки всех справочных статей, для которых нет ссылок типа род – вид, часть – целое. [2]

Алфавитный словарь включает в себя перечень наименований всех информационных статей в алфавитном порядке. [2]

К основным элементам гипертекстовой технологии относятся: [3]

- информационный фрагмент;

- тема;

- узлы;

- ссылки.

Информационный фрагмент гипертекста может представлять собой линейную последовательность строк текста, рисунок, видеофрагмент, аудиофрагмент.[5]

Тема содержит краткое название информационного фрагмента. Информационный фрагмент может состоять целиком из множества тем либо включать в себя одну или несколько тем наряду с прочей информацией.

Узлом в гипертексте называется информационный фрагмент, из которого возможен переход к другим информационным фрагментам гипертекста.[5]

Ссылка представляет собой слово, фразу или набор фраз, с помощью которых осуществляется переход от одного узла к другому. Ссылки могут быть референтными или организационными.

Референтные ссылки — это наиболее типичный вид ссылок в гипертекстах. Они, как правило, имеют два конца, обычно это направленные связи, хотя большинство гипертекстовых информационных систем поддерживает и обратное движение по

ссылке. Исходный конец референтной ссылки называется «источник». Логически это отдельная точка или область в тексте. Другой конец называется «назначением» — это определенная точка или область в гипертексте. С источником ссылки связывается некоторая пометка, указывающая наличие ссылки, — она показывает имя ссылки, обычно изображается в виде последовательности символов и высвечивается как отдельная единица текста. Например, при щелчке по термину появится информационный фрагмент, разъясняющий значение этого термина.[5]

Организационные ссылки устанавливают явные связи между двумя точками гипертекста и отличаются от референтных тем, что поддерживают иерархическую структуру в гипертексте. Организационные ссылки связывают узел-родитель с узлами-сыновьями и, таким образом, формируют древовидный подграф в рамках общего гипертекстового сетевого подграфа. Такие ссылки часто соответствуют отношению «быть частным случаем», и по этой причине операции над этими ссылками (при построении гипертекста) отличаются от операций над референтными ссылками.[6]

Область применения гипертекстовой технология очень широка. Это издательская деятельность, библиотечная работа, обучающие системы, разработка документации, законов, справочных руководств, баз данных, баз знаний и т.д. [6]

Наиболее известным примером гипертекста являются веб-страницы — документы HTML (язык разметки гипертекста) как они размещаются в Сети.

Современные программы разработки Web-серверов, такие как MS FrontPage или Web Pen для Windows, дают возможность даже новичку без всякого штудирования учебников легко создавать готовые странички. При этом специалист по созданию Web-сайтов, называемый Web-мастером, берет готовые файлы (тексты, таблицы, графику, базы данных, звук, анимацию, видеофильмы, программы) и с помощью кнопок и команд меню оформляет страницы сайта. Подобные программы, выполняя команды инструментальных и операционного меню, формируют гипертекст WWW-сервера.

Исходные текстовые, табличные и графические и другие объекты включаются в Web-site посредством тегов (tag = ярлык, этикетка). Тег - это последовательность символов, задающая

1). положение объекта на странице сайта,

2). внешний вид объекта или

3). связь данной страницы с другими страницами этого сайта, а также с любым другим сервером.

Тег называют также управляющим маркером, флагом. Программы типа Web Pen сами расставляют теги, поэтому пользователь таких программ может не знать языка разметки гипертекста (HTML = HyperText Markup Language).

1.2 История возникновения гипертекстовой разметки

История гипертекста богата и переменчива, поскольку гипертекст не столько какая-то новая идея, сколько находящаяся в эволюции концепция возможного применения компьютера. В разработку идеи гипертекста внесли свой вклад много людей, и каждый из них, видимо, представлял себе нечто отличное от других.

Компьютерному гипертексту предшествует ручной, один из вариантов которого – традиционное использование карточек. Такие карточки можно нумеровать и снабжать взаимными ссылками. Их часто распределяют по рубрикам, т. е. им придается иерархическая организация (в некотором ящике или пакете). Удобство таких карточек состоит в том, что, имея небольшой размер, они разбивают записи на малые куски. Пользователь может легко реорганизовать картотеку с учетом новой информации. Но, конечно, с увеличением объема такой картотеки, работать с ней становится все труднее.[13]

Другой вариант ручного гипертекста – это справочная книга, например словарь и энциклопедия. Статьи или определения, даваемые в таких книгах, содержат явные ссылки друг на друга, последовав за этими ссылками, читатель получает более богатую информацию. Каждой такой книге можно поставить в соответствие сеть с текстовыми узлами и связями-ссылками.[13]

Многие века существуют документы, где внутренние перекрестные ссылки и отсылки к другим документам образуют значительную долю содержания. Таковы, например, Талмуд с его обильным использованием аннотаций и встроенным в текст комментарием, а также сочинения Аристотеля, в которых ссылки на другие источники играют огромную роль.[13]

Еще один важный пример – печатные издания Библии. В них текст каждой из ее книг-частей делится на главы, а те, в свою очередь, на стихи. Главы пронумерованы внутри каждой книги, стихи – внутри каждой главы. Стих может состоять из части грамматического предложения, одного целого предложения или нескольких фраз . В подлинном библейском тексте этого деления нет. Оно было сделано учеными-богословами для облегчения ссылок и цитат. К примеру, деление Нового Завета на стихи, ныне общепринятое, восходит к XVI веку. Согласно этому делению, Новый Завет (27 книг-частей) состоит из 260 глав и, суммарно по всем главам, из 7942 стихов. Гипертекст возникает здесь потому, что в современных изданиях Библии текст идет в сопровождении так называемых "параллельных мест", обычно в виде ссылок на полях. Каждая такая ссылка ставит в соответствие стиху, который идет рядом, "параллельные" стихи из этой же или других книг-частей Библии (даются координаты этих стихов). В комментариях объясняется, что "параллельные места" указывают на тождественные события и "созвучные выражения". Библейский текст, по существу, превращен в гипертекстовую сеть на узлах-стихах. Сеть имеет огромные размеры: если взять лишь ее новозаветную часть с ее внутренними "параллелями", то получилось бы почти 8 тыс. гипертекстовых узлов (из них, правда, многие не имели бы связей).[7]

Все эти примеры относят появление гипертекста к далеким временам. Сейчас, однако, немало специалистов, которые считают, что об истинном гипертексте можно говорить лишь в том случае, когда перемещение по связям поддерживается компьютером.

В 1945 году в своей статье “Как мы можем думать” (“As We May Think”) Ванневар Буш высказал идею машины для просмотра и пополнения записями документов, записанных на пленке. [7]

Эта машина, получившая название “Memex”, никогда не была построена, но она содержала идею, которую позже назвали гипертекстом. [7]

Информация, считал Буш, должна храниться в виде пленочных микрофильмов. Поэтому Memex имела устройство для чтения микрофильмов и устройство для записи микрофильмов с помощью процесса сухой фотографии. То есть Memex принципиально не была цифровым компьютером. [8]

Буш представлял машину в виде письменного стола с экранами для отображения информации и клавиатурой для управления. Внутри стола размещалось хранилище микрофильмов и механизм доступа к ним (рисунок 1)[8]

Рисунок 1 – Схема Memex

Однако главная особенность и новизна Memex состояла не в способах хранения информации, а в способе доступа к ней. Буш предложил механизм перекрестных ссылок, аналогичный тому, который используется в современном гипертексте. Ссылки Буш предлагал записывать во вспомогательных полях в теле документа, а в качестве средства навигации использовать нечто похожее на современный мультимедийный шлем.[8]

Фактически система Xanadu явилась прообразом web-пространства, но сам Тед Нельсон отзывается отрицательно о Web и HTML, считая, что работа Бернерса-Ли является сильно упрощенным вариантом его работы. [8]

Первую работающую гипертекстовую систему (она называлась NLS, от *oN Line System*) продемонстрировал в 1968 году Дуг Энгельбарт (Doug Engelbart, 1925 г.р.). [8]

В 1989 году Тим Бернерс-Ли (Tim Berners-Lee, 1955 г.р.), работая над внутренней сетью организации CERN (Европейский совет по ядерным исследованиям в Женеве), предложил глобальный гипертекстовый проект, ныне известный как Всемирная паутина. [7]

В начале 1990-х Тим Бернерс-Ли и его коллеги создали язык HTML, на котором записываются современные гипертексты, то есть такие документы (распределенные по Сети), которые благодаря гиперссылкам можно просматривать по контексту. После создания языка HTML Web-технологии начали приобретать более четкий характер и бурное разностороннее развитие, поддерживаемое многочисленными компаниями в будущем, такими как: Google, Microsoft, Mozilla Foundation и пр. Историческая последовательность изложена в приложении А.[8]

1.3 Модели гипертекста

В самом общем виде модель гипертекста характеризуется механизмом связей, узлами-объектами и пользовательским интерфейсом – способом взаимодействия человека с узлами и связями.

Узлы. Узел – важнейшее понятие гипертекстовых систем, так как в них именно в форме узлов хранится и представляется пользователю информация. Характеристики узла, существенные для гипертекстового пользователя (читателя или автора), – это тип информации, которая может быть сохранена (текст, таблица, графика, звук и др.), и вместимость, объем каждого узла.[13]

Некоторые гипертекстовые системы поддерживают только текст (например, NLS и ZOG), другие – таблицы и графику (HyperTIES, HyperCard, Guide); есть механизмы интерпретации различных видов информации (например, видео) в рамках гипертекстовой сети (Intermedia, NoteCards).

Важно отметить, что просто реализация возможности показывать узлы мультимедиа здесь недостаточна. Каждый новый тип информации должен быть полностью интегрирован с гипертекстовой сетью системы, для чего необходимо разработать методы создания связей между, например, кадрами видео и текстом. В Intermedia и NoteCards это достигается сравнительно легко: они разрабатывались в расширяемых средах (объектно-ориентированный язык Си и Lisp соответственно), отсюда – легкая встраиваемость в систему модулей, создающих гипертекстовую функциональность для новых видов информации.

Что касается объемов узлов, то, например, такие системы, как ZOG и HyperCard, поддерживают только узлы жестко фиксированной величины (объемом с экран), в то время как другие обладают более гибкими возможностями.

Узел на экране обычно дается в своем отдельном окне. Одновременно может быть открыто лишь несколько окон-узлов. Выполняются стандартные операции оконных систем. Отметим, что сами оконные системы, а также манипулятор мышь, были изобретены Дагласом Энгельбартом, для нужд его пионерской системы.[13]

Объем узла, разбиение информации на узлы – серьезная проблема для автора гипертекста, который должен думать о восприятии смыслового содержания узла читателем. Предпочитительны узлы, обладающие внутренним смысловым единством (внутренней когерентностью).[13]

Связи. Способ реализации связей имеет ключевое значение, поскольку именно связи обеспечивают "нелинейное ветвление" – сердцевину гипертекстовой

функциональности.[13]

При описании связей в гипертексте часто используется понятие anchor (буквально, якорь) – это слово или фраза, которые подсвечиваются на экране и воспринимаются как точки начала или конца связи.

Возможны два варианта статуса гипертекстовых связей. В первом связи являются самостоятельными объектами, которыми пользователь может манипулировать напрямую; во втором связи спрятаны в системе (возможно, как часть текста) и проявляются, только когда пользователь каким-либо образом задействует их.[13]

Intermedia – пример системы со связями первого статуса. У нее связи хранятся отдельно от документов, на которые ссылаются. Эти связи также могут быть типизированы с помощью данных вида атрибут – значение, что позволяет осуществлять их поиск по запросу пользователя.

Связи этого статуса делают возможным включение в систему графического браузера – средства, с помощью которого сеть (граф из узлов и связей) отображается на экране. Без хранения межузловых связей невозможно говорить о графовой структуре сети. Поэтому лишь такие системы, как NoteCards, Intermedia и реализации HAM (Hypertext Absnract Machine), могут обеспечивать функциональность, необходимую для обработки сети как графа.

Связи второго статуса ("спрятанные") – есть просто спецификации адреса для перехода и существуют только в момент их активизации. Например, связи в HyperCard – это кнопки, содержащие инструкцию "иди к карточке № 42106" (или что-то подобное). Однако кнопки могут и не содержать таких инструкций, они будут целиком вставлены в программный код системы. Так как нет ясного соотношения между узлом и набором его связей в гипертекстовой сети, HyperCard не имеет средств для воздействия на сеть как целое. Хотя система и обеспечивает графическое представление последних посещенных карточек, не поддерживается информация об альтернативных маршрутах, по которым можно следовать.[13]

В системах NLS, HyperTIES, ZOG адрес узла для перехода по связи хранится как часть текста или имя отдельно стоящей кнопки. Такие связи по своей природе однонаправленные, они позволяют осуществлять сквозной проход по документу, но при этом возможно оказаться в тупике (в узле, из которого не выходят связи).[13]

Еще одной важной характеристикой связи является то, как она подключена к узлам – соединяет ли она узлы как целые или фрагментами. Еще по поводу Memex

отмечалось, что, когда узлы соединяются как целые, у читателя могут возникать трудности с пониманием причины существования конкретной связи от данного узла, особенно если у него эта связь не единственная. В свою очередь, в узле, куда произошел переход, читатель часто вынужден искать информацию, которая делает переход осмысленным.

Наиболее гибко эти проблемы были решены в Intermedia. В этой системе связи могут начинаться из любого фрагмента одного узла и заканчиваться в любом фрагменте другого. В Notecards и HyperCard, как и Memex, связь относится целиком к узлу-карточке. У NLS, ZOG, HyperTIES и Guide началом связи всегда служат некоторые слова и фразы.[13]

Интерфейс пользователя. Эволюция гипертекстовых систем тесно переплетена с исследованиями, разработками, изобретениями в области человеко-машинного интерфейса. Иконки, кнопки, оконные интерфейсы, манипуляторы типа мышь и многое другое изобрели люди из гипертекстового сообщества. И выдающимся изобретателем в этой области является пионер гипертекста Даглас Энгельбарт. Он жив до сих пор и очень почитаем в компьютерном мире. Несколько лет тому назад была учреждена премия Энгельбарта, вручаемая на ежегодной международной конференции серии Hypertext за лучшую представленную работу.[13]

А влияние гипертекстовых разработок на развитие новых идей в области человеко-машинного интерфейса видно по тому факту, что программы регулярных всемирных конференций по взаимодействию компьютера с человеком обязательно включают раздел, посвященный гипертексту.

Однако вернемся к различным вариантам реализации интерфейса пользователя в обозреваемых гипертекстовых системах. Эти варианты с их особенностями отражают поиски и пути, которыми шли исследователи.

Все системы позволяли легко перемещаться по документу в его естественной последовательности – либо путем скроллирования линейного документа (Intermedia, Guide), либо двигаясь по дереву иерархической структуры узлов, используя операции "следующий потомок" или "возврат к родителю" (ZOG, HyperCard).

Значительно больший интерес представляют вопросы: как пользователь может распознать связь (каково ее визуальное представление) – и как эту связь можно активизировать? В разных системах эти вопросы решаются по-разному. В Memex информация о связях просто хранилась на специально отведенном пустом

пространстве узла как сигнал читателю: что-то из данного текста связано с дополнительной информацией. Все остальные системы позволяли размечать информацию узла так, чтобы его части могли стать метками связей: у одних (ZOG, HyperTIES и Guide) это была подсветка соответствующего ключевого слова или фразы, у других (HyperCard, Intermedia, Notecards) – значки связей или кнопки, которые, как значки сноски (примечания), предупреждали читателя о существовании дополнительной информации, соотнесенной с данной.

Возможность выбора той или иной связи достигалась, например, в Memex и ZOG соотнесением каждой данной связи с ключом клавиатуры (так, как это делается для меню). В Intermedia, Notecards, HyperCard и Guide вместо этого надо было указать на связь и "кликнуть" мышью. У HyperTIES немного другая модель (клавиатурная): одна связь-точка остается подсвеченной и эта подсветка перемещается от одной связи к другой под управлением курсорных стрелок. Когда нужная связь выбрана, пользователь активизирует ее с помощью некоторого другого ключа.

В NLS подобная операция также состоит из двух частей – выбрать и активизировать. Такой подход позволяет управлять как выбором, так и последующим действием (так называемый подход "selections and actions").[14]

Когда связь распознана и выбрана, система осуществляет немедленный гипертекстовый прыжок по связи к новой информации. Такой мгновенный прыжок больше подходит компьютеру, чем человеку. Читатель, например при просмотре журнала, встретив ссылку (или цитату) на другую книгу или статью, не бросает чтение, а доводит его до конца и лишь потом по библиотечной ссылке смотрит работу, на которую ссылались. В случае с компьютером такие переходы по связям ведут к поиску в глубину с растущей стопкой (стеком) отложенных статей, к которым надо будет вернуться. Такие отложенные статьи могут расти, как снежный ком, увеличивая нагрузку на память и отвлекая внимание читателя. При этом многие системы могли высвечивать только один узел на экране в данный момент.

При нескольких переходах по связям у пользователя терялся контекст исходной информации. Он забывал, с чего начал свои переходы и где находится, особенно когда приходилось исследовать неизвестную сеть. Это существенный недостаток гипертекста – быстрая дезориентация пользователя.

Второй побочный эффект и недостаток гипертекста, называемый когнитивной перегрузкой, связан с необходимостью совершать множество действий (выбирать связи, кнопки, совершать переходы, возвращаться назад) для получения полезной информации. В гипертекстовых системах были разработаны средства борьбы с этими недостатками. Например, Notecards и Intermedia позволяли многим узлам находиться на экране одновременно, в Intermedia разрабатывались графические браузеры, отображающие в отдельном окне структуру сети связей (локальные и глобальные карты). При этом существовала опасность утопить пользователя во множестве открытых окон с разнообразной информацией.[14]

Таким образом, введение в обычный текстовый формат гипертекстовых (перекрестных) связей увеличивает функциональность системы по сравнением со случаем статично-линейного текста. Однако некоторые особенности гипертекстовых систем делают написание гипертекста тяжелым трудом , а сам гипертекст – трудным для восприятия.

Преимуществам нелинейности, перекрестных переходов и мультимедийной информации – всему тому, что есть "больше, чем текст", угрожают недостатки, описанные выше. В некоторых обстоятельствах гипертекст, вероятно, является менее подходящим, чем линейный текст.

Пути и навигация. Борьба с недостатками гипертекста насчитывает не один десяток лет, и хотя полностью их устранить нельзя, многое сделано в этом направлении.[13]

Прежде всего, в гипертексте изначально существует метафора прокладывания пользователем пути (тропы, трейла) в паутине гипертекстовых связей. Путь – это последовательность из узлов и связей, которые посещает пользователь. Это понятие выработал еще Ванневар Буш, имея в виду аналогию с процессами в мозгу человека: "Человеческий ум работает ассоциативно. Ухватив, поняв что-то, он сразу цепляется за следующее, что предлагается, подсказывается ассоциацией мыслей в соответствии с некоторой сложной паутиной трейлов, которые поддерживаются ячейками мозга".

Обычно гипертекстовые системы кроме возможности переходов по связям предоставляют и различные методы, помогающие прокладывать пути, т. е. осуществлять навигацию в гипертексте. Эти методы, не ограничивая свободу пользователя, направлены на преодоление дезориентации и дополнительной когнитивной нагрузки, от которых страдает читатель гипертекста и которые

вместе получили название "проблема навигации" (Navigation Problem).

У большинства гипертекстовых систем навигационную помощь предоставляют следующие средства.

Локальная карта. Это картинка всех связей и узлов, непосредственно связанных с текущим узлом. Она может быть графической (например, в виде блок-схемы) или текстовой (просто список). Локальные карты обеспечивают читателю контекст и помогают выбрать связь.[13]

Глобальная карта. Это графическое представление полной сети из узлов и связей. Ввиду трудностей с отображением огромного числа связей, такие карты мало пригодны для реальных гипертекстов объемом свыше сотни узлов. Больше всех с ними экспериментировали разработчики Intermedia (Янкелович, Мейровиц, ван Дам). Локальные и глобальные карты в реальных гипертекстовых системах назывались "графическими браузерами".[13]

История (бэктрекинг). Посещенные узлы и связи текущего трейла сохраняются и есть возможность вернуться в предыдущие узлы.[13]

Туры (проложенные маршруты). Это хранящиеся пути, которые можно проходить по желанию. Подобное полезно при создании гипертекстовых учебников или демонстраций. Для больших сетей значение туров возрастает.[13]

Поиск (в совокупности гипертекстовых узлов). Используются все достижения в области информационного поиска: булевские запросы, морфологический поиск, языки запросов и др.[13]

Фильтры. Это возможность ограничения области навигации пределами задаваемого фильтром подмножества узлов и связей. Такие подмножества называются видами (views) и могут быть сохранены для последующего повторного доступа.[13]

Индекс. Список подсвеченных слов, связей или узлов, упорядоченных по алфавиту, теме, автору, предмету и т. д. Индексы разрабатываются автором и имеют тот недостаток, что никак не учитывают точку зрения читателя на то, как он хочет использовать гипертекст.[13]

Закладки. Читатель может сохранить (пометить) свою текущую позицию, чтобы вернуться к ней позже.[13]

Для больших и сложных гипертекстовых сетей растет необходимость использовать гибкую, интеллектуальную помощь в навигации.

В заключение – об одном таком подходе, принадлежащем отечественным исследователям и направленном на преодоление проблемы навигации в гипертексте.

В этом подходе пользователь и система ведут в сети когерентную навигацию, подразумевается, что навигационная тропа должна быть подобна когерентному дискурсу (тексту, предложения которого вместе образуют смысловое единство). Это означает, что кроме локальной связности между парами узлов в тропе-трейле должна поддерживаться и некая глобальная связность, подчиненность разворачиванию некоторой темы, заданной начальным узлом тропы.[5]

Такая когерентная навигация, конечно, нужна не всегда. Например, человек, осуществляющий навигацию в гипертексте, может хаотически перемещаться по связям в надежде попасть на интересные факты, получить ответ на какой-то частный вопрос, когда найденные факты рассматриваются по отдельности, вне контекста.[6]

Однако есть и альтернативные виды человеческой активности при взаимодействии с гипертекстом, где такая навигация нужна. Это – браузинг по определенной тематике, или изучение какого-то предмета по материалу, собранному в сети, или же подготовка чернового варианта документа из узлов сети.[6]

Когерентная навигация реализована в системе СМИСК – российской разработке. В этой системе локальные переходы по связям в строящейся тропе-трейле происходят под так называемым макроконтролем, следящем за глобальной (тематической) связностью. Этот контроль базируется на иерархии подтем строящегося дискурса.

Иерархия реализована в форме интерактивного дерева, которое развертывается в тропу-дискурс.

Теоретически разработки СМИСК опираются на труды таких известных психолого-лингвистов, как ван Дийк, Кинч и Левельт. Результаты были представлены в докладе на гипертекстовой секции международной конференции "Восток-Запад" EWHCI'93.

Глава 2. Виды языков гипертекстовой разметки

Самый популярный на сегодняшний день язык гипертекстовой разметки HTML, был создан специально для организации информации, распределенной в сети Интернет, и является одной из ключевых составляющих технологии WWW. С использованием гипертекстовой модели документа способ представления разнообразных информационных ресурсов в сети стал более упорядочен, а пользователи получили удобный механизм поиска и просмотра нужной информации. [15]

HTML (HyperText Markup Language) - язык гипертекстовой разметки, который в настоящее время используется в World Wide Web. Изначально создавался как язык для обмена научной и технической документацией. Стандартизацией языка HTML занимается W3C (WWW Consortium). [15]

HTML является упрощенной версией стандартного общего языка разметки - SGML (Standart Generalised Markup Language), который был утвержден ISO в качестве стандарта еще в 80-х годах. Этот язык предназначен для создания других языков разметки, он определяет допустимый набор тэгов, их атрибуты и внутреннюю структуру документа. Контроль за правильностью использования дескрипторов осуществляется при помощи специального набора правил, называемых DTD-описаниями, которые используются программой клиента при разборе документа. Для каждого класса документов определяется свой набор правил, описывающих грамматику соответствующего языка разметки. С помощью SGML можно описывать структурированные данные, организовывать информацию, содержащуюся в документах, представлять эту информацию в некотором стандартизованном формате. Но в виду некоторой своей сложности, SGML использовался, в основном, для описания синтаксиса других языков (наиболее известным из которых является HTML), и немногие приложения работали с SGML- документами напрямую.[15]

2.1 SGML

SGML — метаязык, на котором можно определять язык разметки для документов. SGML — наследник разработанного в 1969 году в IBM языка GML (Generalized Markup Language).[16]

Изначально SGML был разработан для совместного использования машинно-читаемых документов в больших правительственных и аэрокосмических проектах. Он широко использовался в печатной и издательской сфере, но его сложность затруднила его широкое распространение для повседневного использования.

Основные части документа SGML: [16]

1. SGML-декларация — определяет, какие символы и ограничители могут появляться в приложении;
2. Document Type Definition — определяет синтаксис конструкций разметки. DTD может включать дополнительные определения, такие, как символьные ссылки-мнемоники;
3. Спецификация семантики, относится к разметке — также даёт ограничения синтаксиса, которые не могут быть выражены внутри DTD;
4. Содержимое SGML-документа — по крайней мере, должен быть корневой элемент.

Язык SGML предоставляет множество вариантов синтаксической разметки для использования различными приложениями. Изменяя SGML-декларацию, можно даже отказаться от использования угловых скобок, хотя этот синтаксис считается стандартным, так называемым *concrete reference syntax*.

Пример синтаксиса SGML: [16]

<quote type="example">

typically something like <italics>this</italics>

</quote>

SGML стандартизован ISO: «ISO 8879:1986 Information processing—Text and office systems—Standard Generalized Markup Language (SGML)».

Языки HTML и XML произошли от SGML. HTML — это приложение SGML, а XML — это подмножество SGML, разработанное для упрощения процесса машинного разбора документа. Другими приложениями SGML являются SGML Docbook (документирование) и «Z Format» (типография и документирование).

2.1.1 Описательная разметка

Система описательной разметки использует коды разметки, просто предоставляющие названия для классификации частей документа. Коды, такие, как <para> или \end{list} просто идентифицируют часть документа и утверждают про нее: "следующий элемент - параграф" или "это - конец начатого последним списка" и т.д. Напротив, система процедурной разметки определяет, какая обработка должна производиться в конкретной точке документа: "здесь вызвать процедуру PARA с параметрами 1, b и x", или "сдвинуть левую границу на 2см влево, правую -- на 2см вправо, пропустить строку и встать на новую левую границу", и т.д. В SGML инструкции, необходимые для обработки документа с определенными целями (например, для его форматирования) четко отделяются от описательной разметки, встречающейся внутри документа. Обычно они собираются вне документа в отдельных процедурах или программах.

При описательной, а не процедурной, разметке один и тот же документ можно обрабатывать различными программами, каждая из которых может применять различные правила обработки к тем частям документам, которые она считает важными. Например, программа анализа содержимого может совершенно игнорировать сноски в аннотируемом тексте, тогда как программа форматирования может извлекать и собирать их вместе для печати в конце каждой главы. С одними и теми же частями файла могут ассоциироваться разные правила обработки. Например, одна программа может выделять имена людей и географические имена для создания индекса или базы данных, а другая, оперирующая тем же текстом, может печатать имена собственные шрифтом отличающегося начертания.

2.1.2 Типы документов

SGML вводит понятие типа документа и, как следствие, определения типа документа (document type definition, DTD). Тип документа формально определяется его составными частями и их структурой. Например, определение отчета может констатировать, что он состоит из заголовка, возможно, автора, за которым следуют аннотация и один или несколько абзацев. Все, что не имеет заголовка, в соответствии с этим формальным определением, отчетом не является, так же, как не является им последовательность абзацев, за которой следует аннотация, вне зависимости от того, насколько такие документы похожи на отчет для читателя-человека.

Раз документы имеют известные типы, можно использовать специальную программу, называющуюся анализатором (parser), для проверки документа, утверждающего свою принадлежность определенному типу. Анализатор проверяет, что все элементы, требуемые типом документа, на самом деле присутствуют и расположены в правильном порядке. Что более важно, разные документы одного и того же типа могут обрабатываться одинаковым образом. Можно конструировать программы, использующие знание структуры документа, которые, таким образом, могут действовать в более осмысленной манере.

2.1.3 Независимость данных

Основная цель создания SGML заключалась в том, чтобы обеспечить транспортабельность закодированных документов из одной аппаратной и программной среды в другую без потери информации. Два описанных выше свойства решают эту задачу на абстрактном уровне; третье свойство -- на уровне строк байтов (символов), из которых составляется документ. SGML предоставляет универсальный механизм строковой подстановки (string substitution), то есть, простой машинно-независимый способ обозначить, что некоторая последовательность символов в документе должна заменяться при его обработке некоторой другой последовательностью. Одно очевидное применение этого механизма -- обеспечение согласованности номенклатуры; другое, и более важное, -- противодействие печально известной неспособности различных компьютерных систем понимать наборы символов друг друга, или способ в любой системе предоставить все графические символы, необходимые для конкретного приложения, путем использования описательных обозначений непереносимых символов. Строки, определенные этим механизмом подстановки, называются объектами (entities). В SGML слово *объект* (*entity*) несет специальный смысл: оно означает именованную часть размеченного документа, безотносительно ко всяческим соображениями структуры. Объектом может быть строка символов или целый файл текста. Для включения его в документ используется конструкция, известная как *ссылка на объект* (*entity reference*).

2.1.4. SGML-структуры

Этот раздел описывает простой и согласованный механизм разметки или идентификации структурных единиц текста, предоставляемый SGML. Он также

описывает, какие способы SGML предлагает для выражения правил, определяющих возможные осмысленные комбинации этих единиц в любых текстах.

Элементы

В стандарте SGML для текстовых единиц, рассматриваемых как структурные компоненты, используется термин ***элемент*** (*element*). Различным типам элементов даются различные названия, но SGML не предлагает никаких способов выразить значение конкретного типа элементов, кроме его отношения к другим типам элементов. То есть, все, что можно сказать про элемент, называющийся (например) **<blort>**, -- это то, что его экземпляры могут встречаться (а могут и не встречаться) внутри элементов типа **<farble>**, и что он может раскладываться (а может и не раскладываться) на элементы типа **<blortette>**. Следует подчеркнуть, что стандарт SGML совершенно не заботит семантика текстовых элементов: она зависит от приложения (В настоящий момент идет работа по созданию (с использованием синтаксиса SGML) определения стандартного"языка семантики и спецификации стилей документов (*document style and semantics specification language*, DSSSL)".) Дело создателей SGML-совместимых наборов разметок (таких, как описанный в этом Руководство) -- выбрать осмысленные имена идентификаторов элементов и документировать правильное их использование в разметке текстов. Это -- одна из целей данного документа. От необходимости выбора названий элементов, кодирующих их функцию, происходит технический термин для названия типа элемента: ***обобщенный идентификатор*** (*generic identifier*), или GI. [16]

В размеченном тексте (***экземпляре документа***, *document instance*) каждый элемент должен быть явно размечен или отмечен некоторым образом. Стандарт предоставляет несколько разных способов это сделать, наиболее часто используемый из них -- вставить метку (*tag*) в начале элемента (***открывающая метка***, *start-tag*) и еще одну -- в конце элемента (***закрывающая метка***, *end-tag*). Пара открывающей и закрывающей меток используется для выделения элементов в тексте, так же, как разные скобки или кавычки используются в обычной пунктуации. Например, элемент цитирования может быть отмечен в тексте так: [16]

реплика Розалинды <quote>Ничего глупее я никогда не

слышала!</quote> ясно показывает ...

Как показывает данный пример, открывающая метка имеет вид **<название>**, где открывающая угловая скобка означает начало открывающей метки, "название" -- обобщенный идентификатор отмечаемого элемента, и закрывающая угловая скобка означает конец метки. Закрывающая метка имеет аналогичный вид, за исключением того, что за открывающей угловой скобкой стоит символ косой черты, так что соответствующая закрывающая метка будет **</название>**. (На самом деле символы, используемые в качестве ограничителей (угловые скобки, косая черта, восклицательный знак) могут переопределяться, но удобно использовать символы, приведенные в этом описании.) [16]

Модели содержимого элемента: пример

Элемент может быть ***пустым*** (*empty*), то есть, не содержать внутри вообще ничего; элемент может содержать просто текст. Чаще, однако, элементы одного типа будут целиком содержаться (*embed*) внутри элементов другого типа. [16]

Возможность использования правил, устанавливающих, какие элементы могут быть вложены в другие, является очень важным свойством SGML. Не переходя к дальнейшему разбору этих правил, можно попытаться рассмотреть, как размеченный вышеприведенным образом текст может быть обработан с различными целями. Простая индексирующая программа может выделять только значимые элементы текста для генерации списка заголовков, или слов, использованных в тексте стихотворения; простая программа форматирования может вставлять пустые строчки между строфами, возможно, начиная с красной строки первую строчку каждой строфы, или вставляя номер строфы. Разные части каждого стихотворения могут набираться разными способами. Более сложная аналитическая программа может соотносить использование знаков пунктуации со строфовыми и метрическими разделами. Исследователи, желающие видеть следствия изменений разделов строф или строк, выбранных редактором этого стихотворения, могут это сделать просто меняя положения меток. И, конечно, представленный выше текст может быть перенесен с одного компьютера на другой и обработан любой программой (или человеком), понимающей смысл внесенных в него меток, безо всяких преобразований и трансляций, необходимых обычно для перемещения файлов текстовых процессоров. [16]

Определение структуры документов SGML: DTD

Правила наподобие вышеописанных -- первый шаг в создании формальной спецификации структуры SGML документа или ***определения типа документа***,

обычно сокращаемого как ***DTD***. При создании DTD дизайнер документа может задавать произвольно жесткую или сколь угодно гибкую структуру. Нужно найти компромисс между удобством следования простым правилам и сложностью поддержки реальных текстов. Это особенно справедливо, когда определяемые правила относятся к уже существующим текстам: дизайнер может иметь очень туманное представление об изначальном предназначении или смысле старых текстов, и задание непротиворечивых правил, касающихся их структуры, может быть очень сложным. С другой стороны, когда специфицируется новый текст, например, для ввода в некоторую текстовую базу данных, то чем точнее установлены правила, тем лучше они могут быть выдержаны. Даже в случае разметки уже существующего текста может иметь смысл определить ограничивающий набор правил, относящихся к определенному видению текста или гипотезе, касающейся текста, -- хотя бы как средство проверки полезности этого видения или гипотезы. Важно помнить, что каждое определение типа документа является интерпретацией текста. Не существует единственного DTD, охватывающего все сведения о тексте, хотя может быть удобно предпочитать одни DTD другим для конкретных типов анализа. [16]

В настоящее время SGML шире всего применяется там, где основным требованием является единообразие структуры документов. Например, при производстве технической документации весьма важно, чтобы разделы и подразделы были соответствующим образом вложены, чтобы перекрестные ссылки были корректны, и так далее. В таких ситуациях к документам относятся как с сырому материалу, к которому применяется заранее определенный набор правил. Однако, как говорилось выше, использование простых правил может также сильно упростить задачу аккуратной разметки элементов и менее ограниченных текстов. Делая такие правила явными, исследователь уменьшает свою работу по разметке и проверке электронного текста, в то же время выявляя интерпретацию структуры и значимые особенности кодируемого текста. [16]

Правила минимизации

Вторая часть описания задает ***правила минимизации*** для элемента. Эти правила определяют, обязаны ли присутствовать открывающая и закрывающая метки для каждого появления данного элемента. Они имеют вид пары символов, разделенных пробелом, первый из которых относится к открывающей, а второй -- к закрывающей метке. В обоих случаях должны присутствовать или минус или буква O; минус означает, что метка должна присутствовать, а буква O -- что она может быть опущена. Так, в нашем примере каждый элемент, кроме **<line>**, должен

иметь открывающую метку. Только элементы **<poem>** и **<anthology>** обязаны также иметь и закрывающую метку. [16]

Модель содержимого

Третья часть каждого описания, заключенная в круглые скобки, называется ***моделью содержимого*** элемента, потому что она указывает, что могут содержать экземпляры элемента. Содержимое указывается либо в терминах других элементов, либо при помощи специальных зарезервированных слов. Есть несколько таких зарезервированных слов, из которых самое часто используемое -- #PCDATA. Это сокращение от *parsed character data* (разобранные символьные данные), и оно означает, что описываемый элемент может включать любые разрешенные символьные данные. Если представить себе SGML описание в виде структуры наподобие генеалогического дерева, с одним предком наверху (в нашем случае, это будет **<anthology>**), то почти всегда, если следовать по ветвям дерева вниз (например, от **<anthology>** к **<poem>**, **<stanza>**, **<line>** или **<title>**), мы придем к #PCDATA. В нашем примере так определены **<title>** и **<line>**. Так как в их модели содержимого указано только #PCDATA и не названо никаких включаемых элементов, то они не могут содержать другие элементы. [16]

Обозначения включения

Вышеприведенное описание для **<stanza>** устанавливает, что строфа состоит из одной или более строк. Оно использует ***обозначение включения*** (*occurence indicator*) -- знак плюс -- для указания того, сколько раз может встречаться элемент, поименованный в модели содержимого. В синтаксисе SGML есть три обозначения включения, обычно представленных знаком плюс, вопросительным знаком и звездочкой. (Так же, как и ограничители, эти знаки имеют формальные наименования и могут быть переопределены соответствующим SGML описанием.) Знак плюс означает, что соответствующий элемент может встречаться один или более раз; вопросительный знак означает, что может быть не более одного элемента; звездочка означает, что элемент может или отсутствовать, или появляться один и более раз. Так, если бы модель содержимого для **<stanza>** была (LINE*), были бы допустимы строфы без строк, так же, как и с более чем одной строкой. Если бы она была (LINE?), то пустые строфы были бы тоже допустимы, но ни одна строфа не могла бы иметь более чем одну строку. Описание **<poem>** в примере устанавливает, что **<poem>** не может иметь больше одного заголовка (но может не иметь ни одного) и что оно должно иметь как минимум одну **<stanza>** (и может иметь несколько). [16]

Связки

Модель содержимого (TITLE?, STANZA+) содержит больше одного компонента. Поэтому нужно дополнительно указать порядок, в котором эти элементы (**<title>** и **<stanza>**) могут появляться. Это упорядочение определяется ***связкой*** (*group connector*) -- запятой -- использованным между ее компонентами. Существуют три возможных связки, обычно представляемых запятой, вертикальной чертой и знаком "&". (Так же, как ограничители и обозначения включения, связки имеют в стандарте формальные имена и могут быть переопределены соответствующим SGML описанием.) [16]

Запятая означает, что оба компонента, которые она соединяет, должны встречаться в порядке, указанном в модели содержимого. Знак "&" указывает, что компоненты, которые он соединяет, должны встречаться оба, но в произвольном порядке. Вертикальная черта означает, что может встречаться только один из компонентов, которые она соединяет. Если бы в нашем примере запятую заменить на знак "&", то заголовок мог бы появляться или перед строфами стихотворения, или в его конце (но не между строфами). Если ее заменить на вертикальную черту, то стихотворение могло бы состоять или из заголовка, или только из строф -- но не из того и другого.

Группы модели

До сих пор в нашем примере компоненты каждой модели содержимого были или единственным элементом, или #PCDATA. Вполне можно, однако, определять модели содержимого, в которых компонентами являются списки элементов, объединенные связками. Такие списки, известные как ***группы модели*** (*model groups*), могут также модифицироваться обозначениями включения и, в свою очередь, быть объединенными связками. Чтобы продемонстрировать эти возможности, расширим наш пример так, чтобы включать нестрофовые формы стихов. Для демонстрации классифицируем стихотворения на ***строфовые*** (*stanzic*), ***двустишия*** (*couplets*), и ***белые*** (*blank*) или ***??*** (*stichic*). Белый стих состоит просто из строк (игнорируем пока возможность стихотворных абзацев) Двустишие определяется как **<line1>**, за которой идет **<line2>**. [16]

<!ELEMENT couplet O O (line1, line2) >

Элементы **<line1>** и **<line2>** (которые различаются, например, чтобы сделать возможными изучение схемы рифмования) имеют в точности ту же модель содержимого, что и существующий элемент **<line>**. Они, следовательно, могут

разделять одно и то же описание. В этой ситуации удобно указать ***группу названий*** (*name group*) в качестве первого компонента единого описания элемента, а не записывать последовательность описаний, отличающихся только используемыми именами. Группа названий -- это список GI, соединенный связками и заключенный в круглые скобки: [16]

<!ELEMENT (line | line1 | line2) O O (#PCDATA) >

Описание элемента **<poem>** теперь можно изменить так, чтобы включить все три варианта:

<!ELEMENT poem - O (title?, (stanza+ | couplet+ | line+)) >

То есть, стихотворение состоит из необязательного заголовка, за которым следует одна или несколько строф, либо одно или несколько двустиший, либо одна или несколько строк. Отметьте разницу между этим определением и следующим:

<!ELEMENT poem - O (title?, (stanza | couplet | line)+) >

Второй вариант, используя обозначение включения у группы, а не у каждого элемента внутри группы, позволит одному стихотворению состоять из смеси строф, двустиший или белого стиха.

Таким образом можно строить довольно сложные модели, отражая структурную сложность различных типов текстов. В следующем примере мы рассмотрим строфовый стих, в котором появляется рефрен (*refrain*). Он может состоять из повторений элементов-строк или быть просто текстом, не разделенным на стихотворные строки. Рефрен может появляться только в начале стихотворения или как необязательное дополнение после каждой строфы. Это можно выразить моделью содержимого наподобие следующей:

<!ELEMENT refrain - - (#PCDATA | line+)>

<!ELEMENT poem - O (title?,

((line+)

| (refrain?, (stanza, refrain?)+))) >

То есть, стихотворение состоит из необязательного заголовка, за которым следует или последовательность строк, или безымянная группа, открывающаяся рефреном,

за которым идет одна или несколько других групп, каждый член которой состоит из строфы с необязательным рефреном. Этому образцу отвечает последовательность *рефрен - строфа - строфа - рефрен*, так же, как и *строфа - рефрен - строфа - рефрен*. А последовательность *рефрен - рефрен - строфа - строфа* ему не удовлетворяет, так же, как и *строфа - рефрен - рефрен - строфа*. Среди прочих условий, накладываемых этой моделью, -- требования, чтобы в стихотворении было хотя бы одна строфа, если оно не состоит просто из строк, и чтобы при наличии и заголовка и строфы они появлялись именно в этом порядке.

2.2 HTML

HTML — стандартный язык разметки документов во Всемирной паутине. Большинство веб-страниц содержат описание разметки на языке HTML (или XHTML). Язык HTML интерпретируется браузерами; полученный в результате интерпретации форматированный текст отображается на экране монитора компьютера или мобильного устройства. [15]

Язык HTML является приложением SGML (стандартного обобщённого языка разметки) и соответствует международному стандарту ISO 8879.[15]

Язык HTML был разработан британским учёным Тимом Бернерсом-Ли приблизительно в 1986—1991 годах в стенах ЦЕРНа в Женеве в Швейцарии. HTML создавался как язык для обмена научной и технической документацией, пригодный для использования людьми, не являющимися специалистами в области вёрстки. HTML успешно справлялся с проблемой сложности SGML путём определения небольшого набора структурных и семантических элементов — дескрипторов. Дескрипторы также часто называют «тегами». С помощью HTML можно легко создать относительно простой, но красиво оформленный документ. Помимо упрощения структуры документа, в HTML внесена поддержка гипертекста. Мультимедийные возможности были добавлены позже.[15]

Изначально язык HTML был задуман и создан как средство структурирования и форматирования документов без их привязки к средствам воспроизведения (отображения). В идеале, текст с разметкой HTML должен был без стилистических и структурных искажений воспроизводиться на оборудовании с различной технической оснащённостью (цветной экран современного компьютера, монохромный экран органайзера, ограниченный по размерам экран мобильного телефона или устройства и программы голосового воспроизведения текстов).

Однако современное применение HTML очень далеко от его изначальной задачи. Например, тег <table> предназначен для создания в документах таблиц, но часто используется и для оформления размещения элементов на странице. С течением времени основная идея платформонезависимости языка HTML была принесена в жертву современным потребностям в мультимедийном и графическом

2.2.1 Структура HTML-документа

HTML — это теговый язык разметки документов, то есть любой документ на языке HTML представляет собой набор элементов, причем начало и конец каждого элемента обозначается специальными пометками, называемыми тегами. Регистр, в котором набрано имя тега, в HTML значения не имеет. Элементы могут быть *пустыми*, то есть не содержащими никакого текста и других данных (например, тег перевода строки **
). В этом случае обычно не указывается закрывающий тег. Кроме того, элементы могут иметь *атрибуты*, определяющие какие-либо их свойства (например, размер шрифта для тега **<font>). Атрибуты указываются в открывающем теге. Вот пример части разметки HTML-документа:

Текст между двумя тегами - открывающим и закрывающим.

Здесь элемент содержит атрибут href.

А вот пример пустого элемента:

Каждый HTML-документ, отвечающий спецификации HTML какой-либо версии, обязан начинаться со строки декларации версии HTML <!DOCTYPE>, которая обычно выглядит примерно так:

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"*http://www.w3.org/TR/html4/strict.dtd*"> [15]

Если эта строка не указана, то добиться корректного отображения документа в браузере становится труднее.

Далее обозначается начало и конец документа тегами <html> и </html> соответственно. Внутри этих тегов должны находиться теги заголовка (<head></head>) и тела (<body></body>) документа.

2.2.2 Основные элементы

Теги и их параметры нечувствительны к регистру. То есть <A HREF="http://example.com"> и <a href="http://example.com"> означают одно и то же.

В последних версиях HTML практически у каждого тега огромное число необязательных параметров — обычно не меньше 15. Приведем основные.

Гиперссылки [15]

название ссылки

- Атрибут href задает значение адреса документа, на который указывает ссылка.
- filename — имя файла или адрес Internet, на который необходимо сослаться.
- название ссылки — название гипертекстовой ссылки, которое будет отображаться в браузере, то есть показываться тем, кто зашел на страницу.
- target — задает значение окна или фрейма, в котором будет открыт документ, на который указывает ссылка. Возможные значения атрибута:
    - _top — открытие документа в текущем окне;
    - _blank — открытие документа в новом окне;
    - _self — открытие документа в текущем фрейме;
    - _parent — открытие документа в родительском фрейме.

Значение по умолчанию: _self.

Тот же элемент используется для создания так называемых «якорей» (anchor), которые могут потом использоваться в гиперссылках, направленных на какой-то определённый элемент страницы. Например:

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>Якорь внутри документа</title>

</head>

<body>

* *

* текст *

* Наверх *

</body>

</html>

Аналогичным образом якорь можно сделать на закладку, находящуюся на другой веб-странице или на другом сайте: там, куда направлена ссылка, должен стоять , а там, откуда идёт ссылка, к значению href добавляется знак решётки и название якоря.

Текстовые блоки [15]

- <H1> … </H1>, <H2> … </H2>, … ,<H6> … </H6> — заголовки 1, 2, … 6 уровня. Используются для выделения частей текста (заголовок 1 — самый крупный, 6 — самый мелкий).
- <P> — новый абзац. Можно в конце абзаца поставить </P>, но это не обязательно.
-
 — новая строка. Этот тег не закрывается (то есть не существует тега </BR>)
- <HR> — горизонтальная линия
- <BLOCKQUOTE> … </BLOCKQUOTE> — цитата. Обычно текст сдвигается вправо.
- <PRE> … </PRE> — режим preview (preformatted text). В этом режиме текст заключается в рамку и никак не форматируется (то есть теги, кроме </PRE>, игнорируются, и переводы строки ставятся там, и только там, где они есть в оригинальном документе).
- <DIV> … </DIV> — блок (обычно используется для применения стилей CSS)
- <SPAN> … </SPAN> — строка (обычно используется для применения стилей CSS)

Форматирование текста [15]

- <EM> … </EM> — логическое ударение (обычно отображается *курсивным шрифтом*)
- <STRONG> … </STRONG> — усиленное логическое ударение (обычно отображается жирным шрифтом)
- <I> … </I> — выделение текста *курсивом*
- <B> … </B> — выделение текста жирным шрифтом
- <U> … </U> — подчёркивание текста
- <S> … </S> (или <STRIKE> … </STRIKE>)— ~~зачёркивание~~ текста
- <BIG> … </BIG> — увеличение шрифта
- <SMALL> … </SMALL> — уменьшение шрифта
- <BLINK> … </BLINK> — мигающий текст. Внимание! Этот тег не работает в браузере Internet Explorer версий 5 и ниже без применения JavaScript
- <MARQUEE> … </MARQUEE> — сдвигающийся по экрану текст.
- _… — подстрочный текст. Например, H₂O создаст текст H_2O.
- […] — надстрочный текст. Например, E=mc² создаст текст $E=mc^2$.
- <FONT *параметры*> … </FONT> — задание параметров шрифта. У этого тега есть следующие параметры:
  - COLOR=*цвет* — задание цвета. Цвет может быть задан в шестнадцатеричной форме как #rrggbb (первые 2 шестнадцатеричные цифры задают красную компоненту, следующие 2 — зелёную, последние 2 — синюю) или названием.
  - FACE=*шрифт* задание гарнитуры шрифта
  - SIZE=*размер* задание размера шрифта. Размер от 1 до 7: стандартный по умолчанию 3. Есть много способов изменить стандартный размер.
  - SIZE=+*изменение* или SIZE=-*изменение* — изменение размера шрифта от стандартного. Например, +2 означает размер на 2 больше стандартного.

Списки[15]

<UL>

<LI> первый элемент </LI>

<LI> второй элемент </LI>

<LI> третий элемент </LI>

</UL>

создаёт список

- первый элемент
- второй элемент
- третий элемент

Если вместо <UL> (*Unordered List* — ненумерованный список) поставить <OL> (*Ordered List* — нумерованный список), список получится нумерованным:

1. первый элемент
2. второй элемент
3. третий элемент

У этих тегов есть параметры:

type = "тип"

где *тип* — форма: в <UL> — символов

1. square — квадрат
2. circle — окружность
3. disk — круг: по умолчанию

а в <OL> — цифр или букв

- A или a (латинскими буквами) — буквенный список: соответственно заглавными или строчными буквами
- I или i — римские цифры: соответственно заглавными или строчными буквами

Объекты [15]

- EMBED — вставка различных объектов: не-HTML документов и media-файлов
- APPLET — вставка Java-апплетов
- SCRIPT — вставка скриптов.

Изображения [15]

- IMG — вставка изображения. Этот тег не закрывается.
    - SRC — имя или URL
    - ALT — альтернативное имя (отобразится, если в браузере запретить отображать картинки)

- TITLE — краткое описание изображения (отобразится при наведении курсора на картинку)
- WIDTH, HEIGHT — размеры (если не совпадают с истинными размерами картинки, то изображение «растянется» или «сожмется»)
- ALIGN — задает параметры обтекания текстом (top, middle, bottom, left, right)
- VSPACE, HSPACE — задают размеры вертикального и горизонтального пространства вокруг изображения

Пример:

<IMG SRC=url ALT="текст" TITLE="текст" WIDTH="размер (пикс, %)" HEIGHT="размер (пикс, %)">

Изображение можно сделать ссылкой:

<A HREF=url ><IMG SRC=url></A>

2.3 XML

XML — рекомендованный Консорциумом Всемирной паутины язык разметки, фактически представляющий собой свод общих синтаксических правил. XML предназначен 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*. XML является упрощённым подмножеством языка SGML.[17]

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 (*namespace*). [17]

2.3.1 Достоинства XML

- XML(человеко-ориентированный) — это формат, одновременно понятный и человеку и компьютеру;
- XML поддерживает Юникод;
- в формате XML могут быть описаны основные структуры данных — такие 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 парсингу, что позволяет ему оставаться простым, эффективным и непротиворечивым.
- XML также широко используется для хранения и обработки документов как он-лайн, так и офф-лайн:
- XML — формат, основанный на международных стандартах;
- иерархическая структура XML подходит для описания практически любых типов документов;
- 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 на строке

2.3.2 Недостатки XML

- 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.

    - 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 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
    - Выражение не иерархических данных (например, графов) требует дополнительных усилий
    - Кристофер Дейт отмечал, что «…XML является попыткой заново изобрести иерархические базы данных…» (в 1980-е года иерархические базы данных были вытеснены реляционными базами данных).
- Пространства имён XML сложно использовать и их сложно реализовывать в XML парсерах
- Существуют другие, обладающие сходными с XML возможностями, текстовые форматы данных, которые обладают более высоким удобством чтения человеком (YAML, JSON, SweetXML). Также в последнее время очень большое распространение получил формат fb2.

2.3.3 Принцип построения XML-документа

В общем случае XML-документы должны удовлетворять следующим требованиям: [17]

1. В заголовке документа помещается объявление XML, в котором указывается язык разметки документа, номер его версии и дополнительная информация
2. Каждый открывающий тэг, определяющий некоторую область данных в документе обязательно должен иметь своего закрывающего "напарника", т.е., в отличие от HTML, нельзя опускать закрывающие тэги
3. В XML учитывается регистр символов
4. Все значения атрибутов, используемых в определении тэгов, должны быть заключены в кавычки
5. Вложенность тэгов в XML строго контролируется, поэтому необходимо следить за порядком следования открывающих и закрывающих тэгов
6. Вся информация, располагающаяся между начальным и конечными тэгами, рассматривается в XML как данные и поэтому учитываются все символы форматирования (т.е. пробелы, переводы строк, табуляции не игнорируются,

как в HTML)

Конструкции языка [17]

Содержимое XML-документа представляет собой набор элементов, секций CDATA, директив анализатора, комментариев, спецсимволов, текстовых данных. Общая схема представлена на рисунке 2.

Пример XML-документа:

<conservatory>

<flower>rose</flower>

<flower>tulip</flower>

<flower>cactus</flower>

</conservatory>

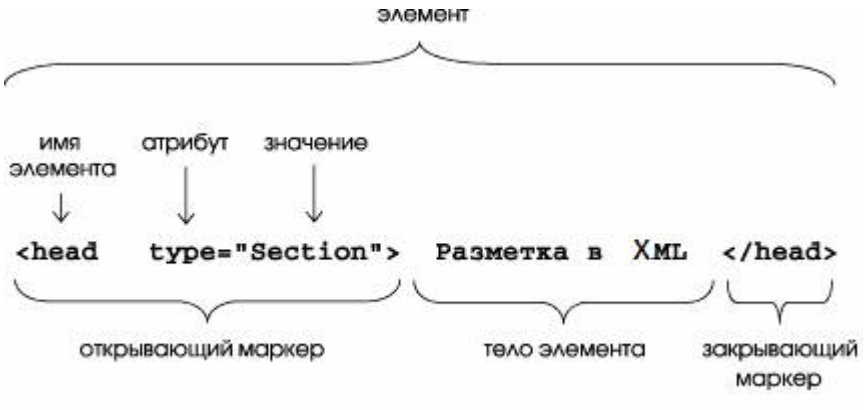

Рисунок 2- Общая схема структуры XML-документа

Элементы [17]

Элемент - это структурная единица XML- документа. Заключая слово rose в в тэги <flower> </flower> , мы определяем непустой элемент, называемый <flower>, содержимым которого является *rose*. В общем случае в качестве содержимого элементов могут выступать как просто какой-то текст, так и другие, вложенные, элементы документа, секции CDATA, инструкции по обработке, комментарии, - т.е. практически любые части XML- документа.

Любой непустой элемент должен состоять из начального, конечного тэгов и данных, между ними заключенных. Например, следующие фрагменты будут

являться элементами:

<flower>rose</flower>

<city>Novosibirsk</city>

Набором всех элементов, содержащихся в документе, задается его структура, и определяются все иерархическое соотношения. Плоская модель данных превращается с использованием элементов в сложную иерархическую систему с множеством возможных связей между элементами.

Производя в последствии поиск в этом документе, программа клиента будет опираться на информацию, заложенную в его структуру - используя элементы документа. То есть, если, например, требуется найти нужный университет в нужном городе, используя приведенный фрагмент документа, то необходимо будет просмотреть содержимое конкретного элемента <university>, находящегося внутри конкретного элемента <city>. Поиск при этом, естественно, будет гораздо более эффективен, чем нахождение нужной последовательности по всему документу.

В XML документе, как правило, определяется хотя бы один элемент, называемый корневым и с него программы-анализаторы начинают просмотр документа. В приведенном примере этим элементом является <country>

В некоторых случаях тэги могут изменять и уточнять семантику тех или иных фрагментов документа, по разному определяя одну и ту же информацию и тем самым предоставляя приложению-анализатору этого документа сведения о контексте использования описываемых данных. Например, прочитав фрагмент **<city>**Holliwood**</city>** мы можем догадаться, что речь в этой части документа идет о городе, а вот во фрагменте **<restaurant>**Holliwood**</restaurant>** - о забегаловке.

В случае, если элемент не имеет содержимого, то есть нет данных, которые он должен определять, то он называется пустым. Примером пустых элементов в HTML могут служить такие тэги HTML, как **
 <hr>, <img>**. Необходимо только помнить, что начальный и конечные тэги пустого элемента как бы объединяется в один, и надо обязательно ставить косую черту перед закрывающей угловой скобкой (например, **<empty/>**;)

Имена тегов и атрибутов можно писать и по-русски. Опыт HTML показал, сколь важна тщательная и своевременная интернационализация всех аспектов языка, претендующего на какую-то роль в Интернете. Поэтому создатели XML позаботились, в частности, о том, чтобы в именах тегов и атрибутов можно было пользоваться не только латинскими буквами, но и кириллицей, иероглифами и вообще всеми символами из репертуара Unicode, которые считаются "буквами" хотя бы в одном языке или системе письменности.

Секция CDATA используется для того, чтобы обозначить части документа, которые не должны восприниматься как разметка. Секция CDATA начинается со строки '<![CDATA[' и заканчивается строкой ']]>'. Внутри самой секции не должна присутствовать строка ']]>'.

Секция CDATA:

<example> <![CDATA[<aaa>bb&cc<<<]]> </example>

Структура XML-документа и разбор его XML-процессором позволяют произвести только простую проверку того, что документ является правильно оформленным. Для создания на этой основе специализированных языков необходимы дополнительные средства описания этих языков. XML поддерживает два механизма подобных описаний: *определения типа документа* (document type definition, DTD) и *XML-схемы* (XML schema).

Глава 3. Будущее гипертекстовой разметки

В настоящий момент актуальным стеком технологий для верстки сайтов являются языки HTML5 и CSS3.

3.1 Язык гипертекстовой разметки HTML5

HTML5 – это пятая версия языка HTML.

HTML5 вводит несколько новых элементов и атрибутов, которые отражают типичное использование разметки на современных веб-сайтах. Некоторые из них — семантические замены для использования универсальных блочных ( ) и строчных () элементов, например, <nav> (блок навигации по сайту), <footer> (обычно относится к нижней части страницы или последней строке HTML

кода) или <audio> и <video> вместо <object>. Некоторые устаревшие элементы, которые можно было использовать в HTML 4.01, были исключены, включая чисто оформительские элементы, такие как <font> и , чьи эффекты выполняются с помощью каскадных таблиц стилей. Также в поведении веб снова заострено внимание на важности скриптов DOM (например, Javascript).[18]

Синтаксис HTML5 больше не базируется на SGML, несмотря на подобие его разметки. Однако он был разработан обратно совместимым с обычным парсингом более старых версий HTML. В HTML5 применяется новая вводная строка, которая выглядит как объявление типа документа в SGML, <!DOCTYPE html>, запускающая соответствующий стандартам режим рендеринга. С 5 января 2009 года HTML5 также включает в себя Web Forms 2.0, ранее бывшие отдельной спецификацией WHATWG.

В дополнение к определению разметки HTML5 устанавливает API, который может быть использован с JavaScript. Возможности DOM расширены и фактически используемые свойства задокументированы. Также добавлены новые API, например:[18]

- элемент холст для непосредственного метода рисования в 2D. См. спецификацию Canvas 2D API Specification 1.0;
- контроль над проигрыванием медиафайлов, который может использоваться, например, для синхронизации субтитров с видео[34];
- хранение данных в браузере;
- редактирование документа: загрузка на страницу через выбор (тег <input type="file">) или перетаскиванием (Drag-and-drop)
- Drag-and-drop: предоставляет набор событий для каждого элемента DOM, таких как появление и нахождение в его зоне, благодаря которым разработчик может информировать пользователя о необходимых действиях и идентификаторе перетаскиваемого файла, содержащего адрес, имя, тип, размер и дату изменения;
- управление историей браузера;
- тип MIME и регистрация обработчика протокола;
- микроданные.

Не все выше перечисленные технологии включены в спецификацию W3C HTML5, хотя они есть в спецификации WHATWG HTML. Немного связанных технологий, которые не являются частью ни одной из спецификаций, следуют далее. W3C публикует спецификации для них отдельно:[18]

- геолокация;
- база данных SQL для Web, внутренняя база данных (больше не поддерживаемая);
- Индексированная база данных (IndexedDB) API, индексирование по типу ключ-значение (прежде — WebSimpleDB);
- Файл API, дескриптор обновления файлов и управления ими;
- Работа с системой. Этот API предназначен для того, чтобы обеспечить хранение информации со стороны клиента без управления базами данных;
- Запись в файл, использование API для записи в файл информации из приложения·

HTML5 – на данный момент это уже стандарт в веб-разработке. В дополнение к нему применяются каскадные таблицы стилей третьего поколения – CSS3, о которых пойдет речь в следующем подразделе.

3.2 Каскадные таблицы стилей

Каскадные таблицы стилей являются удобным дополнением, позволяющим производить разметку сайта быстро, качественно и красиво.

CSS используется создателями веб-страниц для задания цветов, шрифтов, расположения отдельных блоков и других аспектов представления внешнего вида этих веб-страниц. Основной целью разработки CSS являлось разделение описания логической структуры веб-страницы (которое производится с помощью HTML или других языков разметки) от описания внешнего вида этой веб-страницы (которое теперь производится с помощью формального языка CSS). Такое разделение может увеличить доступность документа, предоставить большую гибкость и возможность управления его представлением, а также уменьшить сложность и повторяемость в структурном содержимом. Кроме того, CSS позволяет представлять один и тот же документ в различных стилях или методах вывода, таких как экранное представление, печатное представление, чтение голосом (специальным голосовым браузером или программой чтения с экрана), или при выводе устройствами, использующими шрифт Брайля.

Современное поколение каскадных таблиц стилей – CSS3, обладает огромным множеством возможностей для создания анимированного сайта без использования Javascript.

3.3 Выводы о будущем языков гипертекстовой разметки документов.

Использование стека HTML5+CSS3 для верстки сайтов приобрело большую популярность. Данная комбинация технологий идеально подходит для разметки современных сайтов. Однако консорциум W3C на данном этапе не останавливается. Языки HTML5+CSS3 будут пока и дальше поддерживаться, развиваться, но, скорее всего, только в ближайшие 5-10 лет. Это связано с решением W3C создать новые, «идеальные» языки для разработки сайтов. Следующим поколением станут: HTML6 и CSS4. HTML6 предполагает создание одностраничных веб-приложений без использования технологий Javascript. Соответственно синтаксис данного языка будет значительно отличаться от предыдущего поколения – HTML5. Язык CSS4 будет создан с учетом синтаксиса нового HTML6. Но пока оба языки – HTML6 и CSS4 ещё в активной разработке и будут доступны для массового применения нескоро. Хотя, у W3C уже имеются готовые заготовки, которые веб-разработчики могут опробовать уже сейчас, но не все браузеры будут их поддерживать.[19]

Заключение

В ходе курсовой работы был проведен всесторонний анализ языков гипертекстовой разметки документов. Были решены следующие задачи:

1) проведен анализ понятия «гипертекст»;

2) проанализирована история развития гипертекста;

3) проведен анализ моделей гипертекста;

4) изучены виды языков гипертекстовой разметки документов – синтаксис, структуру документов, основные элементы.

5) провести анализ будущего Web-программирования.

Таким образом, цель данной курсовой работы «обзор языков гипертекстовой разметки» была достигнута

Библиография

1. Бройдо В.Л. Вычислительные системы, сети и телекоммуникации СПб, Питер 2012- 464 с.
2. Информатика /под редакцией С.В.Симоновича. СПб, Питер 2011- 400 с.
3. Кирмайер М. Информационные технологии. СПб.: Питер, 2013 – 443 с.
4. Мэтьюз Дж. Web – сервер. СПб.: Символ, 2008 – 356 с.
5. Олифер В. Г., Олифер Н.А. Компьютерные сети. СПб.: Питер, 2007 – 864 с
6. Олифер В. Г., Олифер Н.А. Сетевые операционные системы. СПб.: Питер, 2009 – 539 с.

7) Сайт «HyperText», What is HyperText. CERN. Проверено 20 октября 2015. URL: http://info.cern.ch/hypertext/WWW/WhatIs.html

8) Тед Нельсон. Curriculum Vitae: Theodor Holm Nelson, PhD (англ.). Сайт Теда Нельсона. Проверено 20 октября 2015. URL: http://hyperland.com/TNvita

9) Юлия Шатилова. Какой была бы альтернативная Сеть? Грезы о цифровой вселенной знаний (рус.)(недоступная ссылка — история) (13 августа 2012 года, 16:02). Проверено 20 октября 2015. Архивировано из первоисточника 25 августа 2012.

10) Тед Нельсон. What's On My Mind (англ.). Сайт проекта Xanadu (Тед Нельсон — автор проекта). Проверено 20 октября 2015. URL: http://www.xanadu.com.au/ted/zigzag/xybrap.html

11) Ted Nelson. Literary Machines. — Edition 87.1. — 2007.

12) "Complex information processing: a file structure for the complex, the changing and the indeterminate" in Association for Computing Machinery: Proceedings of the 20th National Conference. Ed. Lewis Winner: 84-100, Cleveland (Canada): ACM. DOI:10.1145/800197.806036

13) Дуванов А.А., История гипертекста // Информатика – 1 сентября. - 2014. - №4. - С.23-24.

14) Костов Д.А., История гипертекста // Эврика – 2013. - №7. – С. 56-60

15) Квинт И.. HTML, XTML и CSS. СПб.: Питер, 2011 – 382 с.

16) Брайн М. SGML and HTML Explained. Addison Wesley, 1997 – с. 584

17) Холзнер С. XML Энциклопедия. Спб.: Питер, 2010 – с. 1092

18) Сухов К. HTML5. Путеводитель по технологии.

19) Сайт консорциума W3C. URL: https://lists.w3.org/Archives/Public/public-whatwg-archive/2015Mar/0071.html

Приложение А

Таблица 1 - Историческая последовательность

Год	**Событие**
1945 г.	выход статьи Ванневара Буша "As We May Think".
1963 г.	Энгельбарт публикует "A Conceptual Framework for the Augmentation of Man's Intellect".
1965 г.	Нельсон вводит термин "гипертекст".
1967 г.	в Брауновском Университете Анди ван Дам и Нельсон разрабатывают Hypertext Editing System (HES), за которой последовал выпуск FRESS в 1968 г.
1968 г.	Энгельбарт демонстрирует NLS на FJCC (Fall Joint Computer Conference), часть проекта Augment, начатого в 1962 г.
1972 г.	начинается разработка ZOG в Университете Карнеги-Мэллона группой, возглавляемой Робертсоном.
1979 г.	Нельсон приступает к проекту Xanadu.

1981 г.	начинается разработка KMS в Knowledge Systems.
1981 г.	Нельсон публикует "Literary Machines", где подробно описан проект Xanadu.
1982 г.	система ZOG установлена на американском атомном авианосце Carl Vinson.
1982 г.	начинается разработка Питером Брауном системы Unix Guide в Университете Кента.
1983 г.	начинается разработка HyperTIES в Университете Мэриленда.
1983 г.	Рэндол Тригг защищает первую диссертацию по гипертексту в Университете Мэриленда.
1984 г.	начинается разработка Notecards в Xerox PARK.
1985 г.	начало разработки Intermedia в Брауновском Университете.
1986 г.	в Университете Северной Каролины приступают к разработке WE (Writing Environment).
1986 г.	Office Workstation Inc. (OWL) выпускает Guide для Макинтоша.
1987 г.	Apple Computers выпускает HyperCard (Бил Аткинсон) – первую гипермедиа авторскую систему, бесплатно устанавливаемую на каждом продаваемом "Макинтоше".
1987 г.	Джефф Конклин публикует свой выдающийся обзор "Hypertext: An Introduction and Survey".

1987 г.	ACM организует первую конференцию по гипертексту Hypertext'87 (Chapel Hill, North Carolina).
1987 г.	выпуск системы Guide для MS Windows.
1989 г.	Шнейдерман и Керсли разрабатывают Hypertext Hands-On! – первую электронную гипертекстовую книгу.
1989 г.	Autodesk, производитель систем CAD, начинает поддержку Xanadu.
1989 г.	Тим Бернерс-Ли выдвигает проект World Wide Web.
1989 г.	коммерческая реализация IRIS Intermedia 3.0.
1989 г.	опубликован "Afternoon, A Story" М. Джойса – первое произведение гипертекстовой беллетристики.
1989 г.	вторая конференция ACM Hypertext'89 (Pittsburgh, Pennsylvania).
1990 г.	первая Европейская конференция по гипертексту ECHT'90.
1990 г.	декабрь основание Научно-технического центра гиперинформационных технологий (ГНТЦ "Гинтех") Министерства связи РФ.
1991 г.	WWW в ЦЕРНе становится первым глобальным гипертекстом.
1991 г.	разработана первая версия пакета ГиперМетод (для DOS) в Ленинградском электротехническом институте.

1991–1999 гг.	проводятся международные гипертекстовые конференции.
1991–1996 гг.	серия конференций Восток-Запад "Взаимодействие человека с компьютером" (EWHCI).
1992 г.	Autodesk отказывается от работ по проекту Xanadu.
1992 г.	создается группа новостей alt.hypertext.
1993 г.	Международная конференция по гипермедиа и гипертекстовым стандартам в Амстердаме.
1993 г.	NCSA (The National Center for Supercomputing Applications) выпускает Mosaic 1.0 (Marc Andreeson, Eric Bina).
1993 г.	первая конференция, посвященная World Wide Web в Женеве.
1994 г.	информационный поток в WWW (Web-трафик) впервые превышает другие трафики Интернет.

Таблица составлена по данным с научного портала «Эврика», URL: http://evrika.tsi.lv/index.php?name=site&page=51