

Содержание:

Введение

Глобальная сеть – это совокупность компьютеров, расположенных на различных расстояниях друг от друга, и система каналов передачи связи: средств коммуникации (переключения), которая обеспечивает соединение пользовательских коммуникационных систем и обмен данными. Глобальные сети (WideAreaNetworks и WAN) создаются крупными телекоммуникационными компаниями для оказания услуг абонентам. Интернет – это всемирная глобальная компьютерная сеть, которая составлена из различных компьютерных сетей, объединённых соглашениями о способах обмена информацией и единой системой адресации. Интернет использует такие протоколы, как TCP/IP. Они удобны тем, что способны обеспечивать дешёвую возможность надёжно и быстро передавать информацию даже по не очень надёжным линиям связи, и в том числе могут строить программное обеспечение, пригодное для работы на практически любой аппаратуре. Система адресации (URL-адреса) предоставляет уникальные координаты каждому компьютеру и каждому пользователю Интернета, и создаёт возможность взять то, что нужно, и передать туда, куда нужно. В 1994 году началась революция – World Wide Web. Всемирная паутина (WWW) соткана из огромного множества Web-страниц, содержащих в себе различную информацию в зависимости от тематики Web-сайта. В основу Web был положен гипертекст (hypertext) – метод связывания блоков, или страниц, данных, придуманный еще в 60-х годах. Однако только в 90-х годах Бернерсом-Ли и его помощниками концепция гипертекста была перенесена в Internet, и создан HTTP – Hypertext Transfer Protocol (протокол передачи гипертекста). С возникновением HTTP родился и World Wide Web. В наше время в Internet существуют миллионы Web-сайтов. Можно получать доступ к информации по различным темам, более того, в Web можно найти и сведения о нем самом и даже о тех технологиях, на которых он основан. Стоит отметить, что данная курсовая работа носит актуальный характер.

Её целью является подробный обзор языков гипертекстовой разметки. Для достижения этой цели требуется решить следующие задачи: 1) провести анализ понятия «гипертекст»; 2) проанализировать всю историю развития гипертекста; 3) провести анализ моделей гипертекста; 4) изучить виды языков гипертекстовой разметки документов.

Глава 1. Гипертекст

1.1 Понятие гипертекста

Гипертекст – это текст со вставленными в него командами разметки, которые ссылаются на любые другие места в этом тексте, другие документы, изображения и т.д. Во время чтения этого текста в специальной программе, (например Notepad++), можно увидеть подсвеченные в тексте слова. Если навести на них курсор и нажать кнопку мышки, то должно высветиться то, на что ссылалось данное слово, допустим, любой другой параграф той же главы текста. В WWW по ключевым словам можно попасть в любой другой текст из другого документа, войти в различные программы, произвести какие-либо действия и т.д. В Internet'е можно получить доступ к чему угодно, (например к E-mail, ftp, Gopher, Archie, USENET News и т.д). В WWW можно ссылаться на данные на других компьютерах в любом месте сети, тогда при переходе на эту ссылку, данные автоматически передадутся на исходный компьютер и можно будет увидеть на экране текст, данные, изображение и даже речь. Это напоминает Gopher, но фактически это принципиально другое. В Gopher имеется жёсткая структура меню, и по ней можно двигаться, как угодно. Структура не зависит от того, что в ней делают и какой документ используют. В WWW «движение» идёт по документу, который имеет какую угодно гипертекстовую структуру. Имея доступ к редактору гипертекстов, можно создать любую структуру рабочей среды, включая программное обеспечение, и это не будет каким-то новым программным обеспечением, а просто гипертекстом. Такая технология – это по сути представление текста в виде многомерной иерархической структуры типа сети. Гипертекст создаётся в результате представлений текста, как ассоциативно связанных блоков информации. Гипертекст серьёзно отличается от обычного (линейного) текста. В таких текстах последовательная структура и она предусматривают чтение слева направо и сверху вниз. Примитивным примером гипертекста является любая энциклопедия или словарь, в каждой их статьях есть отсылки к другим статьям той же энциклопедии. В итоге читать такие тексты можно как угодно: от одной статьи к другой, игнорируя отсылки, либо читать все статьи подряд, сверяясь с отсылками, наконец, можно от одной отсылки переходить к любой другой. Концепция гипертекста весьма проста. Существует база данных, в ней находятся объекты. Как правило, объекты это небольшие текстовые разделы, и посвящены

они тому или иному вопросу. Специальные механизмы дают возможность компьютеру поддерживать ссылки из одних фрагментов в другие текстовые фрагменты. Человек или бот может устанавливать связи между ними. Такая организация получила название "гипертекст". Гипертекст изначально создавался как среда поддерживающая взаимодействие нескольких людей. Создателями он задумывался, как система общественной деятельности. Группа взаимосвязанных сообщений должна была образовывать сеть и поддерживать социальную сеть отношений между авторами коллективного гипертекста.

1.2 История развития гипертекста

Сама идея была придумана Ваннавером Бушем в 1945 году в предложениях по созданию электромеханической информационной системы Memex. Несмотря на то, что В. Буш был советником по науке президента США Рузвельта, идея так и не была реализована.

Термин гипертекст был введен в обращение Тедом Нельсоном в 1965 году для описания документов, выражающих нелинейную структуру идей, в противоположность линейной структуре традиционных книг, фильмов и речи. Более поздний термин "гипермедиа" близок к нему по смыслу, но он отмечает наличие в гипертексте нетекстовых компонентов, таких как анимация, записанные видео и звук.

В 1968 году изобретатель компьютерной мышки Даг Енжильбард продемонстрировал работу с системой, которая имела типичный гипертекстовый интерфейс, и проведена демонстрация была с использованием системы телекоммуникаций. Однако внятно описать свою систему он не смог.

В 1975 году идея гипертекста нашла воплощение в информационной системе внутреннего распорядка атомного авианосца "Carl Vinson", получившая название ZOG. В коммерческом варианте эта система известна как KMS. Работы в этом направлении продолжались и периодически появлялись реализации типа HyperCard фирмы Apple или HyperNode фирмы Xerox. В 1987 была проведена первая специализированная конференция Hypertext'87, материалам которой был посвящён специальный выпуск журнала "Communication ACM".

К 1989 году гипертекст представлял новую, многообещающую технологию, имеющую довольно большое число реализаций с одной стороны, а с другой -

делались попытки построить формальные модели гипертекстовых систем, которые носили описательный характер, и навеянные успехом реляционного подхода описания данных.

Идея Т. Бернерс-Ли заключалась в том, чтобы применить гипертекстовую модель к информационным ресурсам, распределённым в сети, и сделать это нужно было максимально простым способом. В итоге он заложил три краеугольных камня системы из четырёх, существующих по сей день, разработав:

- язык гипертекстовой разметки документов HTML (HyperText Markup Language);
- универсальный способ адресации ресурсов в сети URL (Universal Resource Locator);
- протокол обмена гипертекстовой информацией HTTP (HyperText Transfer Protocol).

Далее команда NCSA добавила к этим трём компонентам четвёртый:

- универсальный интерфейс шлюзов CGI (Common Gateway Interface).

Идея HTML – это пример весьма удачного решения проблемы построения гипертекстовой системы с помощью специального средства управления отображением. На разработку этого языка существенное влияние оказали два фактора: исследования в области интерфейсов гипертекстовых систем и желание обеспечить быстрый и простой способ создания гипертекстовой базы данных, которая была распределена на сети.

В 1989 году стали активно обсуждать проблему интерфейса гипертекстовых систем, или, иначе говоря, способов отображения гипертекстовой информации и навигации в сети. Наиболее удобной формой организации гипертекста были признаны контекстные гипертекстовые ссылки, и также было признано деление на ссылки, связанные со всем документом в целом и отдельными его частями.

Наиболее простым способом создания документа является его набивка в текстовом редакторе. Опыт создания хорошо размеченных для последующего отображения документов в CERN'е был - трудно было найти физика, не пользующегося системой TeX или LaTeX. Также к тому времени уже существовал стандарт языка разметки - Standard Generalised Markup Language (SGML).

1.3 Модели гипертекста

В самом примитивном виде модель гипертекста характеризуется механизмом связей, узлами и пользовательским интерфейсом – способом взаимодействия пользователя с узлами и связями.

1) Узлы

Узел – важное понятие гипертекстовых систем, потому что в них именно в форме узлов представляется пользователю и хранится информация. Характеристики узла, существенные для гипертекстового пользователя – это тип информации, которая может быть сохранена (текст, таблица, графика, звук и прочее), а также вместимость каждого узла.

Редкие гипертекстовые системы могут поддерживать только текст (например, NLS и ZOG), другие – графику и таблицу (HyperTIES, HyperCard, Guide); есть механизмы трактовки разных видов информации (например, видео) в пределах гипертекстовой сети (Intermedia, NoteCards).

Стоит отметить, что просто выполнения возможности показывать узлы мультимедиа здесь не хватает. Каждый новый тип информации должен быть полностью влит в гипертекстовую сеть системы, но для этого следует разработать методы создания связей между, например, текстом и кадрами видео. В Intermedia и NoteCards это выходит довольно легко: они были разработаны в расширяемых средах (объектно-ориентированный язык Си и Lisp соответственно), поэтому имеет место лёгкая встраиваемость в систему модулей, которые создают гипертекстовую функциональность для новых видов информации.

Что до объёмов узлов, то, к примеру, такие системы, как HyperCard и ZOG, поддерживают только узлы жёстко фиксированной величины (объёмом с экран), а другие имеют более гибкие возможности.

Узел на экране, как правило, даётся в отдельном окне. Одновременно может быть открыто только лишь несколько окон. Выполняются типовые операции оконных систем. Стоит отметить, что оконные системы и манипулятор мышью изобрёл Даглас Энгельбарт, для нужд его системы.

Объём узла, разделение информации на узлы – большая проблема для автора гипертекста, так как ему нужно думать о понимании сути узла читателем. Прежде всего, для этих целей подходят узлы, обладающие внутренней когерентностью.

2) Связи

Способ осуществления связей имеет важное значение, поскольку именно они создают "нелинейное ветвление" – основу гипертекстовой функциональности.

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

Существует две разновидности статуса гипертекстовых связей. В первой связи оказываются самостоятельными объектами, которыми пользователь может производить манипуляции напрямую; во второй они спрятаны в системе и появляются, только когда пользователь решает задействовать их.

Intermedia – пример системы со связями первого статуса. Здесь они хранятся отдельно от документов, на которые ссылаются. Такие связи также могут быть типизированы при помощи данных вида «атрибут» – значение, которое допускает осуществлять их поиск по запросу пользователя.

Связи такого статуса позволяют включить в систему графического браузера – средства, при помощи которого сеть отображается на экране. Без хранения межузловых связей нельзя рассуждать о графовой структуре сети. Поэтому только такие системы, как NoteCards, Intermedia и реализации HAM (Hypertext Abstract Machine), обеспечивают функциональность, нужную для обработки сети как графа.

Связи второго статуса ("спрятанные") – по сути, просто спецификации адреса для перехода и имеются только в момент их активизации. Например, связи в HyperCard являются кнопками, содержащими инструкцию "иди к карточке № 53217" (или что-то в этом роде). Но кнопки могут и не иметь таких инструкций, они будут полностью вставлены в программный код системы. Так как нет понятного соотношения между узлом и набором связей в гипертекстовой сети, HyperCard не имеет возможностей для влияния на сеть как целое. Хотя система и даёт графическое представление последних посещённых карточек, у неё отсутствует информация об иных маршрутах, по которым можно следовать.

В системах NLS, HyperTIES, ZOG адрес узла для перехода по связи хранится как часть текста или имя отдельной кнопки. Такие связи по своей сути однонаправленные, и позволяют выполнять сквозной проход по документу, но при таком варианте можно оказаться в тупике (в узле, из которого связи не выходят).

Ещё одной очень важной характеристикой связи является то, каким образом она подключена к узлам – соединяет ли она узлы как целые или только фрагментами.

Ещё по поводу Memex упоминалось, что, когда узлы соединяются как целые, у читателя могут возникать трудности с осознанием причины существования этой связи от данного узла, особенно если у него она не единственная. Ровно как в узле, куда произошёл переход, читатель должен искать информацию, которая делает переход осмысленным.

Наиболее гибко эти проблемы были решены в Intermedia. В данной системе связи имеют возможность начинаться из любого фрагмента одного узла и заканчиваться также в любом фрагменте другого. В Notecards и HyperCard, как и в Memex, связь относится полностью к узлу-карточке. У NLS, ZOG, HyperTIES и Guide началом связи всегда являются некоторые слова и фразы.

3)Интерфейс пользователя

Эволюция гипертекстовых систем тесно связана с исследованиями и разработками в области человеко-машинного интерфейса. Выдающимся изобретателем в этой области является Дуглас Энгельбарт(1925-2013). Несколько лет назад была учреждена премия, названная его именем, вручаемая на ежегодной международной конференции серии Hypertext за лучшую представленную работу.

Но вернёмся к различным вариантам реализации интерфейса. Они показывают поиски и пути, которыми шли изобретатели.

Системы позволяли легко перемещаться по документу в его нормальной последовательности – либо путем «прокрутки» линейного документа (Intermedia, Guide), либо перемещаясь по дереву иерархической структуры узлов, применяя операции "следующий потомок" или "возврат к родителю" (ZOG, HyperCard).

Набольший интерес представляют вопросы: как пользователь может распознать связь, и как её можно активизировать? В разных системах такие вопросы решаются по-разному. В Memex информация о связях хранилась на специально отведенном незаполненном пространстве узла как сигнал пользователю: 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 соответствующей ключевой фразы или слова, у других (HyperCard, Intermedia, Notecards) – значки связей или кнопки, которые предупреждали пользователя о присутствии дополнительной информации, соотнесенной с данной.

Возможность выбора связей достигалась, например, в Memex и ZOG сопоставлением каждой связи с ключом клавиатуры. В Intermedia, Notecards, HyperCard и Guide надо было указать на связь и кликнуть мышкой. У HyperTIES клавиатурная модель, одна связь-точка остается подсвеченной и подсветка перемещается от одной связи к другой, управляемая курсорными стрелками. Когда нужная связь выбрана, пользователь активизирует ее при помощи другого ключа.

В NLS того рода операция тоже состоит из двух частей – выбрать и активизировать. Этот подход разрешает управлять как выбором, так и следующим действием (подход "selections and actions").

Когда связь определена и выбрана, система выполняет молниеносный гипертекстовый прыжок по связи к новой информации. Такой прыжок больше подходит машине, нежели человеку. Читатель при просмотре журнала, встретив ссылку на другую книгу, не бросает чтение, а доводит его до конца и только потом «переходит по ссылке». С компьютером же подобные вещи ведут к накоплению отложенных статей, всё это ведёт к увеличению нагрузки на память. При частых переходах у пользователя терялся контекст изначальной информации. Он забывал, с чего начинал и где в данный момент находится. Это серьёзный недостаток гипертекста – дезориентация пользователя.

Второй побочный эффект и недостаток – когнитивная перегрузка, связан с потребностью осуществлять множество действий для получения нужной информации. В связи с этим, были разработаны средства борьбы с этими недостатками. Notecards и Intermedia позволяли узлам находиться на экране одновременно, в Intermedia создавались графические браузеры, отображающие в отдельном окне структуру сети связей. В итоге, введение в обычный текстовый формат гипертекстовых связей увеличило функциональность системы. Преимуществам нелинейности, перекрестных переходов и мультимедийной информации – всему тому, что есть "больше, чем текст", грозят недостатки, описанные выше. В некоторых случаях гипертекст, возможно, является менее подходящим, чем линейный текст.

Глава 2. Виды языков гипертекстовой разметки

2.1 Язык разметки SGML

SGML (англ. Standard Generalized Markup Language — стандартный обобщённый язык разметки) — метаязык, на котором можно определять язык разметки для документов. SGML — преемник разработанного в 1969 году в IBM языка GML (Generalized Markup Language).

Первоначально SGML разрабатывался для совместного использования машинно-читаемых документов в правительственных и аэрокосмических проектах. Он широко использовался в печатной и издательской сфере, но его сложность затруднила его повсеместное распространение.

Основные части документа SGML:

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

Язык SGML предоставляет множество вариантов синтаксической разметки для использования разными приложениями. Меняя SGML-декларацию, можно даже отказаться от использования угловых скобок, хотя этот синтаксис считается типовым, так называемым Concrete reference syntax.

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. Также приложениями SGML являются SGML Docbook (документирование) и «Z Format» (типография и документирование).

История

Язык разметки документов – это набор инструкций-тегов, которые предназначены для формирования в документах какой-либо структуры и определения отношений между разного рода элементами этой структуры. Теги языка, или управляющие

дескрипторы, кодируются в этих документах каким-либо образом. Они отличаются от основного содержимого документа и служат как инструкции для программы, которая производит показ содержимого документа на стороне клиента. В первых системах для обозначения таких команд использовались символы открывающаяся и закрывающаяся угловая скобки(" < " и " > "), внутри них помещались названия инструкций и их параметры. Теперь такой способ обозначения тегов является стандартным.

В 1969 г. под руководством сотрудника IBM Чарльза Голдфарба был создан первый язык разметки документов Generalized Markup Language (GML). В нём была реализована концепция типа документа и вложенных друг в друга структур. Тип документа - формально конкретного шаблона, который описывает схему внутреннего построения похожих документов. В 1978 г. был разработан SGML (Standard General Markup Language). В его основу был заложен GML. Основанный на языке GML, он позволил перестать использовать конкретные способы представления информации и сосредоточил усилия на продумывании структуры документов с помощью правил определения собственных тегов форматирования, их атрибутов и синтаксиса использования. В рамках SGML была изобретена концепция DTD (Document Type Definition) - определение типа документа. Она допустила возможность связать конкретные синтаксические правила разбора с заданными способами организации структуры документов. В 1989 г. SGML увеличился стандартизированной ISO таблицей стилей DSSSL, которая позволила дополнять описания структуры SGML-документа «оторванными» от платформы способами его представления на экране. В конце 90-х годов быстро набрал популярность новый язык разметки - XML(Extensible Markup Language). Он представляет собой упрощенный вариант SGML и также позволяет создавать пользовательские XML-приложения. Весьма важной оказалась допустимость создания в XHTML-приложении собственных тегов. 26 января 2000 г. была выпущена спецификация нового языка разметки XHTML 1.0. Разработчики, кроме того, получили три DTD-документа для описания типов, определяемых требованиями HTML 4, поэтому теперь можно обрабатывать XHTML-файлы с помощью, по сути, любых XML-анализаторов. В феврале 2002 г. опубликовали рекомендацию стандарта XML 1.1. Стало допустимым использовать произвольные кодировки символов в названиях элементов, удалось решить проблему представления символов перевода строки в различных операционных системах.

Синтаксис

Существуют три характеристики SGML, которые отличают его от других языков:

- его упор на описательную, а не на процедурную разметку;
- его концепция типа документа (document type);
- его независимость от конкретной системы в представлении текста.

Описательная разметка

Система описательной разметки применяет коды разметки, которые предоставляют названия для классификации частей документа. Коды, например, <para> или \end{list} просто распознают часть документа и говорят про нее: "следующий элемент -- параграф" или "это -- конец начатого последним списком" и т.п. А система процедурной разметки решает, какая обработка должна проходить в конкретной точке документа, например: "здесь вызвать процедуру PARA с параметрами 1, d и x". В SGML инструкции, нужные для обработки документа с определёнными целями хорошо отделяются от описательной разметки, которая встречается внутри документа. Как правило, они собираются вне документа в отдельных программах или процедурах.

При описательной, а не процедурной, разметке одни и те же документы можно обрабатывать различными программами, каждая из которых может применять различные правила обработки к важным, по её мнению частям документа. Так, программа анализа содержимого имеет возможность игнорировать сноски в тексте, в то время как программа форматирования может извлекать и собирать их вместе, чтобы напечатать их в конце главы. С одними и теми же частями текста могут происходить разные правила обработки. Например, одна программа может выделять географические имена для создания индекса, а другая, используя тот же текст, может печатать имена своим шрифтом отличающегося начертания.

Типы документов

SGML вводит такое понятие, как тип документа. Он формально определяется своими составными частями. Допустим, определение отчёта может утверждать, что он состоит из заголовка, автора, за ним идут аннотация и несколько абзацев. Всё, что не имеет заголовка, не является отчётом.

Так как у документов имеются известные типы, то можно пользоваться специальной программой, так называемым анализатором (parser). Он проверяет, что все элементы присутствуют и расположены в правильном порядке. Более того, разные документы одного и того же типа могут обрабатываться одинаково.

Атрибуты в SGML. Объекты SGML

В контексте SGML, слово атрибут (attribute), имеет строгий технический смысл и используется для описания информации конкретного появления элемента. Можно добавить атрибут status к экземплярам элементов для обозначения их достоверности, или добавить атрибут identifier, для того, чтобы ссылаться на появление элемента из других мест документа. Атрибуты полезны именно для этих случаев.

Разные элементы могут иметь атрибуты с одинаковыми названиями, но это не мешает им отличаться и они могут иметь различные значения. Если элемент определен имеющим атрибуты, значения атрибутов задаются в документе как пары атрибут-значение внутри открывающей метки экземпляра элемента. Закрывающая метка не может содержать спецификаций атрибут-значение, так как это было бы излишним.

Обсуждавшиеся тут различные свойства SGML имели отношение к разметке структурных элементов документа. SGML также предоставляет простой метод кодирования и наименования частей содержимого документа переносимым образом. В SGML слово объект (entity) имеет определённый 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).

2.2 Язык гипертекстовой разметки HTML

Самым распространённым из языков разметки Web-страниц является HTML. Впервые предложенный в 1974 году Чарльзом Голдфарбом и в дальнейшем после значительной доработки принятый в качестве официального стандарта. Популярность Internet во многом вызвана появлением World Wide Web (WWW), потому что это первая сетевая технология, предоставившая пользователю простой современный интерфейс для доступа к различным сетевым ресурсам. Из-за простоты и удобства применения значительно выросло число пользователей WWW и это привлекло внимание коммерческих структур. При потребности объединить всё множество информационных ресурсов возникла технология, при помощи которой определяется гипертекстовая навигационная система. Той самой технологией и стал язык HTML (HyperText Markup Language, язык разметки гипертекста). Под термином "разметка" подразумевается общая служебная

информация, которая не выводится вместе с документом, но определяет, как должны выглядеть различные фрагменты документа. Например, можно сделать так, чтобы какое-нибудь слово выводилось жирным шрифтом или было подчёркнуто, вывести конкретный абзац особым шрифтом или делать заголовки шрифтом больше основного. На начальном этапе она была очень проста, и почти все пользователи сети одновременно заимели возможность стать создателями и читателями материалов, опубликованных в сети.

На сегодня существует много разных языков разметки. Например, в некоторых программах особая форма разметки определяет смысл каждого пакета из нулей и единиц, отправляемого в Internet. Всё же, любой язык разметки должен решать две основные задачи:

1) язык определяет синтаксис разметки;

2) язык определяет смысл разметки.

Возникновение стандарта SGML было определено необходимостью совместного использования данных разными приложениями и операционными системами. Даже в 60-х годах у пользователей возникали проблемы с совместимостью. Проведя анализ недостатков нестандартных языков разметки, трое учёных из IBM — Чарльз Гольдфарб, Рэй Лори и Эд Мошер сформулировали три главных принципа, которые обеспечивали возможность совместной работы с документами в различных операционных системах.

1) Использование одних и тех же принципов форматирования во всех программах, которые выполняют обработку документов. Наличие единого набора синтаксических конструкций и общей семантики сильно упрощает взаимодействие между программами.

2) Специализация языков форматирования. Имея возможность построения специализированного языка на базе набора стандартных правил, программист не зависит от внешних реализаций и их представлений о потребностях пользователя.

3) Ясное определение формата документа. Правила, которые определяют формат документа, задают количество и маркировку языковых конструкций, которые используются в документе. Применение стандартного формата гарантирует пользователю, что он будет знать структуру содержимого документа.

Эти правила были в основе предтечи SGML — GML (Generalized Markup Language). Разработка GML продолжалась довольно долго, пока в результате соглашения, заключённого международной группой разработчиков, не появился стандарт SGML.

HTML был основан на стандарте SGML гипертекстовый язык разметки документов для их представления в Web. Стандарты языка HTML, разрабатываются и поддерживаются международным консорциумом W3C, основанным Тимом Бернесом-Ли (Tim Berners-Lee). Средства языка обеспечивают форматную разметку документов, также определяют гиперсвязи между документами и/или их фрагментами.

В качестве основы написания кода HTML разработчики выбрали простой текстовый файл. Из этого следует, что гипертекстовая база данных в концепции WWW — набор текстовых файлов, размеченных на языке HTML, определяющий форму представления информации и структуру связей между файлами и другими информационными ресурсами.

Разработчики HTML справились с двумя задачами:

- предоставили дизайнерам гипертекстовых баз данных простое средство создания документов;
- сделали это средство достаточно мощным, чтобы отразить имевшиеся на тот момент представления об интерфейсе пользователя гипертекстовых баз данных.

Первую задачу решили за счёт выбора теговой модели описания документа. Язык HTML позволяет размечать электронный документ, отображающийся на экране с полиграфическим уровнем оформления; результирующий документ содержит различные метки, иллюстрации, и т.п.

Второй серьёзный момент, который повлиял на судьбу HTML, это то, что за основу был взят обычный текстовый файл. Среда редактирования HTML — это нейтральная полоса между простым текстовым файлом и приложением WYSIWYG (what you see is what you get – что вы видите, то и получаете).

Гипертекстовые ссылки, которые устанавливают связи между текстовыми документами, со временем стали объединять различные информационные ресурсы. Система гиперссылок HTML даёт возможность построить систему взаимосвязанных документов по разным критериям. Язык HTML содержит команды, которые позволяют управлять размером и формой шрифтов, размером и расположением

изображений, и так далее. Документы в HTML - формате представляют собой текстовый файл, который содержит в себе все нужные сведения о выводимой на экран информации. Для управления сценариями просмотра страниц сайта можно пользоваться языками программирования, таких как JavaScript, VBScript и Java. Формы для введения данных, подвергающиеся позднее обработке можно обрабатывать с помощью специальных серверных программ (например, на языках PHP или Perl). Язык HTML позволяет пользователю размещать на страницах интерактивные кнопки и гипертекстовые ссылки, соединяющие Web-страницы с другими страницами сайта, равно как и с любыми другими сайтами.

HTML не является языком программирования, как ошибочно думают многие, это лишь язык разметки текста, один из инструментов, который используют при создании Web-страниц.

Статический и динамический HTML-документы

Существует два вида HTML-документов – статические и динамические. Статические документы хранятся в файлах той файловой системы, которая используется web-сервером или браузером при просмотре локальных файлов. При размещении информации на web-сервере можно использовать динамические документы – те, которые не существуют постоянно в виде файлов, а создаются в момент запроса клиента. Причём конечному пользователю не важно, какой из способов используется.

Для создания динамического документа HTML необходима специально написанная программа по правилам, которые определяет web-сервер.

Динамический способ обязывает хранить данные в формализованном виде, например в базе данных.

В тех случаях, когда данные хранятся в формализованном виде, то, используя шаблоны документов, в которых были произведены изменения, генерируются статические документы. Для их генерирования можно использовать любые средства отчётов, которые имеются в той системе управления баз данных (СУБД), в которой обработаны данные.

2.3 Расширяемый язык разметки XML

Вторая половина 90-х годов двадцатого века отметилась серьёзными переменами в технологиях Web. За свою короткую историю, Web приобрёл многие сотни миллионов пользователей во всём мире, в его среде сформированы и поддерживаются огромные информационные ресурсы. Эта глобальная информационная система вторглась во многие области информационных технологий и стала одним из важнейших звеньев инфраструктуры информационного общества.

При этом ограничения, свойственные действующим технологиям Web (Web первого поколения или Web-1), стали сдерживающим фактором дальнейшего его развития. Новые решения в области технологий Web, которые начали воплощаться в жизнь на рубеже веков, направлены на преодоление этих ограничений и создание технологической платформы, обеспечивающей потенциал для появления нового поколения Web (Web-2) и возможностей его развития. Главную роль в технологическом переоснащении Web стал играть разработанный консорциумом W3C новый язык разметки XML (Extensible Markup Language, расширяемый язык разметки). Язык XML – это метаязык, который представляет из себя подъязык SGML.

Стоит заметить, что аббревиатуру XML очень часто используют для обозначения не только самого языка XML, но и других связанных с ним понятий — определяющего язык стандарта W3C, информационных ресурсов XML, комплекса основанных на языке XML стандартов консорциума W3C, которые составляют платформу XML.

Не смотря на то, что язык XML используют в среде Web по прямому назначению — как выразительное средство для представления информационных ресурсов в этой среде, он вместе с тем интенсивно внедряется в другие технологии. Возможности языка и его поддержка механизмами среды Web позволяют использовать XML в качестве языка-посредника для определения форматов обмена данными между всевозможными системами, использующими Internet как коммуникационную среду.

Основная сфера применения стандартов платформы XML — это представление слабоструктурированных данных Web-сайтов в форме XML-документов. Именно для этого и создавался этот язык. Применение XML в этой области позволяет не только представлять в среде Web гиперстраницы в форме XML-документов, но и поддерживать связанные с этими страницами метаданные. С помощью такой технологии создаются такие поисковые машины Web, которые обеспечивают более низкий уровень информационного шума, чем нынешние HTML-технологии.

Практически самой главной целью создания платформы XML является привнесение в среду Web метаданных, которые описывают свойства поддерживаемых в ней информационных ресурсов. Имеется ввиду описание структуры XML-документов и их семантики. При наличии явного описания структуры, их проверку может выполнять браузер. Описание семантики документов может быть полезно для новых или модернизированных поисковых машин, и для различных нуждающихся в нём Web-приложений.

Но довольно часто не учитывается ещё одно важное назначение метаданных, которые описывают информационные ресурсы Web. Они нужны для создания новых высокоуровневых приложений Web, которые основаны на интеграции информационных технологий и обеспечивают интеграцию неоднородных информационных ресурсов.

Пространства имён XML

Самая простая возможность задания семантики — использование пространства имён. В отличие от языка HTML, который обеспечивает форматную разметку текста, определяющую его представление на экране, XML служит для структурной разметки.

Разметка в XML позволяет отмечать в тексте содержательные структурные единицы, которые называют элементами XML-документа. Для выделения каждого типа элементов используется свой собственный тег, который указывает имя типа элемента. Так что с каждым тегом можно ассоциировать семантику соответствующих элементов XML-документа (номера телефонов организаций, адреса и т. п.).

Если сообщество разработчиков XML-документов договорится о единой интерпретации имён, которые принадлежат некоторому набору, то, возможно, этот набор, с каким-либо описанием их смысла может использоваться как пространство имён. Адрес Web-ресурса, символизирующий некоторое пространство имён, может не содержать явного описания смысла принадлежащих ему имён и вообще не существовать. Тут мы имеем дело с определением семантики имён этого пространства по умолчанию.

В последние годы создаются сервисы регистрации и поддержки пространств имён в интересах разных объединений разработчиков и пользователей. Зарегистрированное пространство имён является стандартом для клиентов сервиса регистрации.

Перспективы XML

XML — не модное направление, а естественный результат развития Web-технологий. Серьёзную работу по продвижению стандартов платформы XML ведёт международный, некоммерческий консорциум OASIS (Organization for Structured Information Standards, Организация по стандартизации структурированной информации), в его состав входит более 600 членов со всего мира. Эта деятельность — основная задача консорциума. OASIS контролирует разработки, разрабатывает сам и распространяет информацию о методологиях применения, технологиях и реализациях этих стандартов. Распространению стандартов XML-платформы главным образом содействует политика W3C, которая направлена на создание доступности их спецификаций, создание множества свободно распространяемых синтаксических анализаторов для языка, также огромное внимание уделяется обеспечению преемственности для существующей HTML-платформы.

Язык XML и стандарты, которые на нём основаны, получили широкое распространение. Технологии XML распространились и в нашей стране. Поэтому приобретает большое значение русскоязычная терминология в этой области.

Платформа XML имеет позитивные перспективы для широкого практического применения. Свидетельство этому не только огромные функциональные возможности рассмотренного семейства стандартов, но и высокая активность в области разработки и развития стандартов, а также основанного на них производства программного обеспечения.

2.4 Расширяемый язык разметки XHTML

Динамичное распространение технологий XML создаёт довольно серьёзную проблему обеспечения преемственности в развитии среды Web, создания возможностей, которые позволяют использовать большие информационные ресурсы, накопленные в рамках технологий HTML. Один из подходов к её решению был реализован в стандарте XHTML 1.0 (The Extensible Hypertext Markup Language, расширяемый язык разметки гипертекста), одобренном W3C в январе 2000 года. Как и HTML, XHTML является подмножеством языка SGML, но XHTML, в отличие от предшественника, соответствует спецификации XML.

Идея такого подхода заключалась в создании на основе XML языка разметки, который эквивалентен по его функциональности языку HTML. Таким же образом в настоящее время разработаны множество других языков разметки конкретизации XML.

Авторы стандарта XHTML 1.0 объясняют функцию определяемого в нём языка как переформулировку HTML в XML. XHTML является преемником HTML. Нужда в более строгой версии HTML появилась из-за того, что веб-контент всё больше становится ориентированным на иные виды устройств (планшеты, смартфоны), в которых, как правило, ограничены ресурсы, в том числе и для обработки гибкого, нетребовательного HTML.

Почти все современные браузеры поддерживают XHTML. Он совместим и со старыми браузерами, т. к. в основе XHTML лежит HTML. Серьёзное преимущество XHTML проявляется в сочетании с каскадными таблицами стилей. Такое сочетание допускает отделение оформления документа от его содержимого.

Отличия XHTML от HTML формальны и предназначены только для приведения его в соответствие с XML. Основное требование в том, чтобы все теги были правильно вложены и семантически развиты. Также, в XHTML все теги записываются строчными буквами, все атрибуты должны быть в кавычках. Также все элементы должны быть закрыты, в том числе те, у которых нет закрывающего тега (закрываются добавлением слэша ('/') в конец тега).

В стандарте XHTML предлагается три варианта целевого языка для представления HTML-документов:

- XHTML Strict (строгий XHTML);
- XHTML Transitional (переходный XHTML);
- XHTML Frameset (XHTML с фреймами).

Вариант XHTML Strict полностью отделяет содержание документа от оформления, большинство атрибутов (таких как, bgcolor и align) больше не поддерживаются. Служит только для структурной разметки без применения элементов форматирования. Для целей форматирования можно дополнительно пользоваться языком каскадных таблиц стилей (CSS).

Вариант XHTML Transitional наиболее популярный. Он допускает использование CSS, но будут внесены незначительные изменения в разметку, для того, чтобы старые браузеры могли воспринимать документ.

Вариант XHTML Frameset обеспечивает поддержку фреймов. Что допускает разделение окна браузера на несколько фреймов.

Существуют также и узкоспециальные версии XHTML:

- XHTML 1.1 Модульный (Module-based): в ней можно импортировать дополнительные свойства в разметку. Ещё эта версия поддерживает руби-разметку, нужную для дальневосточных языков;
- XHTML Основной (Basic): более лёгкая версия XHTML для устройств, которые не могут использовать полный набор элементов XHTML (планшеты, телефоны);
- XHTML мобильного профиля (Mobile Profile): основан на XHTML Basic, добавляет специфические элементы для мобильных устройств. Даёт пользователям доступ к полноцветному контенту, он отлично выглядит и имеет удобную навигацию.

В стандарте XHTML много внимания посвящено поддержке развития языка HTML. Считается, что XHTML 1.0 определяет начальную версию развивающегося семейства типов документов, позволяющих воспроизводить HTML, расширять язык и выделять его подмножества.

В спецификации XHTML существует специальное пространство имён XHTML. Но для именованя элементов и атрибутов в документах разрешается использование также и других пространств имён (например, математический язык разметки MathML или Resource Definition Framework (RDF, среда описания ресурса)).

Пространство имен XHTML

XHTML наиболее доступен, потому что использует пространство имён, по сути, идентичное с HTML 4.01 и большая часть Document Type Definition (определение типа документа) уже "зашита" в браузере. DTD очень важен для XML-документов. Ещё один вопрос, который связан с XML: пространство имён этого языка довольно велико и определяется DTD, который написан специально для конкретной его разновидности. Для браузера во время разбора XML-кода эти документы должны быть доступны.

Валидация XHTML документов

Валидным (отвечающим всем правилам) XHTML-документом считается документ, который соответствует технической спецификации. По сути, все браузеры должны следовать веб-стандартам, и такие документы должны отображаться во всех браузерах на всех платформах. Валидация XHTML-документа рекомендована даже вопреки тому, что она не гарантирует кросс-браузерной совместимости. Валидный документ должен содержать определение типа документа (DTD). DTD должен быть расположен до всех других элементов документа. Валидный XHTML-документ, по правилам W3C, может быть отмечен баннером, который подтверждает

правильность XHTML-разметки.

2.5 HTML5

HTML5 — это не продолжатель языка разметки гипертекста, а новая открытая платформа, которая предназначена для создания веб-приложений использующих графику, аудио, видео, анимацию и прочее.

Главная идея разработчиков — сделать жизнеспособный язык разметки, работающий на практике. Один из принципов, положенных в основу стандарта: на первом месте стоят интересы реальных пользователей, затем — создателей сайтов, затем — разработчиков браузеров и других приложений, затем — авторов самого стандарта, и только потом — теоретические и идеологические соображения, которые Консорциум безуспешно продвигал последние десять лет.

HTML5 существует в двух версиях — как разновидность XHTML и как модификация HTML. Создатели веб-страниц применяют любой из этих вариантов. В первом варианте новый язык полностью согласован с XML 1.0, так что разработчикам доступны все преимущества XML и смежных технологий. Во втором — HTML5 покончил с плохим наследием прошлого — формальной совместимостью с SGML . Предыдущие стандарты HTML претендовали на эту совместимость, но ни один браузер не обрабатывал код HTML по непростым правилам SGML. Вместо этого использовали простые алгоритмы, распознававшие набор конкретных элементов и атрибутов.

HTML 5 «легализовал» такое поведение браузеров. Новый стандарт не смотрит на SGML, а описывает частные правила для интерпретации конкретных тэгов и других синтаксических конструкций. В отличие от старых стандартов, которые были предназначены для разработчиков кода, HTML5 строго и подробно описывает поведение браузеров. Он требует от браузеров обрабатывать не только грамотный код, но и корректировать ошибки.

Например, HTML 5 не разрешает авторам кода пользоваться элементами `tbody` или `marquee`, но описывает, как браузеру нужно поступить, если они встретятся в документе. Это соответствует очередному главному принципу HTML5 — максимальная совместимость с уже существующим наполнением сети. Миллионы сайтов не должны пострадать от появления нового языка и поддерживающих его браузеров.

Ещё одна особенность HTML5 заключается в том, что он описывает документ HTML в терминах объектной модели (DOM) . Перед тем, как отобразить веб-страницу, браузер формирует древовидную структуру её элементов. Это позволяет включить в спецификацию объекты, свойства и методы DOM, использующиеся с давних времён: `window.open()`, `alert()`, `prompt()`, `document.links`, `document.images`, `innerHTML`, `contentEditable` и т. д. Также, в HTML 5 официально включён элемент `embed` .

Тем не менее, HTML5 собрал не всё, что поддерживается в браузерах. Некоторые элементы были удалены: `acronym`, `applet`, `basefont`, `big`, `center`, `dir`, `font`, `isindex`, `s`, `strike`, `tt`, `u`. Исчезли фреймы (`frame`, `frameset`, `noframes`), но остался элемент `iframe`. Пропали и атрибуты, которыми мало кто пользовался (`accesskey`, `longdesc` и др.) или которых не было особого смысла (`align`, `border`, `valign`, `cellpadding`, `cellspacing` и пр.). Некоторые элементы было решено переосмыслить: `i` и `b` теперь обозначают разные формы смыслового выделения, а не курсив и полужирный шрифт, как раньше; `hr` отвечает за разрыв в структуре документа. Элемент `menu` теперь позволяет создавать настоящие меню, а `address` «понижен» в правах и предназначен только для контактной информации в узком контексте.

Также было добавлено много нового. Для статей, документации и прочих материалов будут полезны элементы `article` и `section`, а также новая система заголовков и рубрикации. «Шапка» и «подвал», навигационная и боковая панели обозначаются, как `header`, `footer`, `nav` и `aside`. Для мультимедиа появились элементы `audio` и `video`. Для тех браузеров, где такие элементы не поддерживаются, HTML5 позволяет дополнить их похожим содержанием — например, роликом Flash, при помощи которого внедряется большинство аудио- и видеозаписей. В более продвинутых браузерах полноценно работает элемент `audio` или `video`, а во всех остальных — альтернативный ролик, так что все пользователи останутся довольны.

По замыслу разработчиков, новые возможности HTML5 должны создавать как можно меньше проблем в тех браузерах, где они не поддерживаются. Почти все новшества созданы так, что для старых браузеров можно обеспечить их реализацию с помощью альтернативного содержания, CSS, JavaScript и других технологий. Отдельного упоминания заслуживает элемент `canvas`, предоставляющий область для рисования изображений средствами JavaScript. Массовый интерес к HTML 5 начался после того, как Firefox и Opera решили поддерживать этот элемент.

Формы в HTML5 полностью изменились. Примитивные процедуры заполнения и отправки формы из прежних версий HTML, теперь никого не устраивают, поэтому

разработчики сайтов придумывают собственные расширения интерфейса. В HTML5 эта самодеятельность урегулирована стандартными элементами, атрибутами и интерфейсами DOM. В дополнение к обычным текстовым полям, также появились поля для ввода определённых данных. Так, `input type="number"` — это поле для ввода числа, а `input type="range"` — ползунок для регулирования громкости. Атрибуты `min`, `max` и `step` определяют диапазон и точность допустимых значений в поле. Элемент `datalist` обеспечивает поле со списком, а элемент `output` — область для вывода динамически генерируемых результатов. Атрибут `required` для элемента управления обозначает, что данный элемент обязателен для заполнения, так что без него форма не будет отправлена. Атрибут `autofocus` позволяет установить фокус ввода в данный элемент. Метод `checkValidity()` проверяет правильность заполнения полей. До этого такие возможности имитировались на сайтах с помощью кустарных сценариев JavaScript.

В стандарте предлагается элемент `datagrid` для организации динамических структур данных, элемент `command` для вызова пользовательских команд, элемент `progress` — индикатор хода выполнения операции, а также средства DOM для редактирования содержания прямо на веб-странице, перетаскивания, выделения, работы с буфером обмена и многого другого. Это отвечает общей тенденции развития Интернета: если раньше сайты были чем-то вроде публикаций, то теперь они стали полноценными приложениями, требующими функционального и удобного пользовательского интерфейса. Соответственно, новый HTML также должен стать не столько языком публикаций, сколько языком описания приложений.

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;

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

Список использованной литературы

1. Бройдо В.Л. Вычислительные системы, сети и телекоммуникации СПб, Питер 2012- 464 с.
2. Дуванов А.А., История гипертекста // Информатика – 1 сентября. - 2014. - №4. - С.23-24.
3. Информатика /под редакцией С.В.Симоновича. СПб, Питер 2011- 400 с.
4. Квинт И. HTML, XHTML и CSS. СПб.: Питер, 2011 – 382 с.
5. Кирмайер М. Информационные технологии. СПб.: Питер, 2013 – 443 с.
6. Костов Д.А., История гипертекста // Эврика – 2013. - №7. – С. 56-60
7. Мэтьюз Дж. Web – сервер. СПб.: Символ, 2008 – 356 с.
8. Олифер В. Г., Олифер Н.А. Компьютерные сети. СПб.: Питер, 2007 – 864 с
9. Олифер В. Г., Олифер Н.А. Сетевые операционные системы. СПб.: Питер, 2009 – 539 с.
10. Сухов К. HTML5. Путеводитель по технологии.
11. Холзнер С. XML Энциклопедия. Спб.: Питер, 2010 – с. 1092

Интернет ресурсы

1. Сайт консорциума W3C. URL: <https://lists.w3.org/Archives/Public/public-whatwg-archive/2015Mar/0071.html>
2. Сайт «HyperText», What is HyperText. CERN.
URL:<http://info.cern.ch/hypertext/WWW/WhatIs.html>