

Содержание:

Введение

Тема данной курсовой работы — языки гипертекстовой разметки.

Цель исследования — проанализировать особенности и ключевые этапы развития языков гипертекстовой разметки на примере таких языков, как GML, SGML, HTML и XML.

Предмет исследования — языки GML, SGML, HTML и XML.

Этапы решения поставленной цели:

- Определить понятие «Языки гипертекстовой разметки»;
- Выявить основные особенности языков гипертекстовой разметки;
- Изучить историю развития языков гипертекстовой разметки;
- Проанализировать особенности конкретных языков гипертекстовой разметки.

Актуализация языков гипертекстовой разметки как совокупности отдельных идей гипертекста и языков разметки стала возможной только благодаря развитию компьютерных технологий и Всемирной паутины. Они позволили стереть границы коммуникации не только внутри локальных файловых систем, но и между целыми информационными сетями.

Не стоит забывать и про потребности бизнеса, который, особенно в последние годы, оперирует Big Data. Пользователи, посещающие Интернет, находятся под постоянной бомбардировкой персонализированных предложений контекстной рекламы, ориентирующейся на его действиях в Сети — это, отчасти, стало возможным благодаря языкам разметки и их формализованной разметке контента сайта, позволяющей роботам-исследователям эффективно его считывать.

В случае больших онлайн-энциклопедий, вроде Википедии, создание удобных систем гипертекста это уже вопрос не только удобства, но и необходимости — контент множества страниц находится друг с другом в постоянной взаимосвязи, невыразимой без корректного ссылочного аппарата.

1. Определение понятия «Языки гипертекстовой разметки»

Вполне естественно начать курсовую работу с определения объекта исследования, а именно языков гипертекстовой разметки. Основной проблемой нашего исследования является не столько факт недостатка авторитетной литературы, дающей чёткое определение для рассматриваемого понятия, но и буквальное отсутствие его использования. Поэтому в данном исследовании дается специальное (автономное) и последовательное определение языков гипертекстовой разметки.

1.1 Имеющиеся референсы

В большинстве источников под языками гипертекстовой разметки понимается конкретно язык разметки HTML. Действительно, эта аббревиатура так и расшифровывается: HyperText Markup Language, или, с английского языка, «язык гипертекстовой разметки».

Так как объект нашего исследования находится также в названии XHTML, стоит упомянуть и о нем. XHTML, с английского eXtensible HyperText Markup Language, — расширяемый язык гипертекстовой разметки. Его развитие остановлено и новые версии не выпускаются, а W3C (Консорциум Всемирной паутины) рекомендует вместо него использовать HTML. Стоит отметить, что, несмотря на схожие названия, они с HTML кардинально различаются.

Однако, такое ограничение для исследуемого понятия нам кажется избыточным из-за самого определения гипертекста, которое появилось задолго до появления Web'a. Гипертекст — это термин, обозначающий систему из текстовых страниц, имеющих перекрёстные ссылки. В компьютерной терминологии гипертекст — это текст, сформированный с помощью языков разметки с расчетом на использование гиперссылок.

Также рассмотрим понятие гиперссылки. Это часть гипертекстового документа, ссылающаяся на другой элемент в самом документе, на другой объект, расположенный на локальном диске или в компьютерной сети, либо на элементы этого объекта.

Таким образом, применение гипертекста и гиперссылок, даже если рассматривать их исключительно в рамках Web'a, явно не ограничивается использованием HTML или XHTML. Из этого вытекает необходимость в разработке более актуального определения для языков гипертекстовой разметки, по крайней мере, в контексте данной курсовой работы.

1.2 Итоговое определения

Для определения языков гипертекстовой разметки основными являются следующие уже рассмотренные нами термины: гипертекст, гиперссылки, а также языки разметки. Язык разметки в компьютерной терминологии — набор символов или последовательностей, вставляемых в текст для передачи информации о его выводе или строении. Под языком разметки понимают набор соглашений о разметке, применяемых для кодирования текстов. Язык разметки должен специфицировать, какая разметка допустима, какая разметка обязательна, как отличить разметку от текста и что разметка значит.

Таким образом, на основе этих трёх понятий мы можем сформировать определение для языков гипертекстовой разметки — это языки разметки, поддерживающие использование гиперссылок. Стоит также отметить, что языки разметки в данном случае рассматриваются только в контексте компьютерных языков — то есть, языков, относящимся к компьютерной технике.

Уже на основе таких вводных данных мы можем немного расширить предметную область языков гипертекстовой разметки, упомянув попадающие под определение языки разметки — таковыми окажутся, например, GML, SGML, и произошедшие от них HTML и XML.

1.3 Особенности и принципы работы

Как мы уже выяснили, наиболее общее определение для языков разметки, в частности гипертекстовой, звучит так — набор символов или последовательностей, вставляемых в текст для передачи информации о его выводе или строении.

Сформированный с использованием языка (гипертекстовой) разметки текстовый документ содержит не только последовательность букв и служебных символов как текст, но и дополнительную информацию, о его различных участках — это могут

быть различные выделения, списки, заголовки и так далее.

Фактически, такой документ, помимо содержательной части, содержит в себе набор специальных меток-инструкций, называемых тегами (tags), выполняющих определенные функции. Их можно разделить на два подвида, в соответствии с их назначением: логическую и визуальную.

1.4 Логическая функция

Логическая функция (разметка) отвечает за выделение логических элементов и неразрывно связана с семантикой формируемого документа. Она указывает какую именно роль играет данный участок в общей структуре или, в случае некоторых языков разметки, иерархии — например, указание определенной строки текста как заголовка или пункта списка.

1.5 Визуальная функция

Визуальная функция (разметка) отвечает за задание функций обработки выделенных элементов. Она определяет, как именно будет отображаться конкретный элемент — например, позволяет отдельную строку отображать жирным шрифтом.

Стоит отметить, что в случае продвинутых языков (гипертекстовой) разметки визуальная функция может дополняться с помощью отдельных спецификаций, реализуемых иным средствами — об этом чуть ниже.

1.6 Идея языков разметки

Основная идея языков разметки состоит в том, что визуальное отображение документа должно автоматически получаться из логической разметки и не зависеть от его непосредственного содержания. Естественно, что это же актуально и в случае языков гипертекстовой разметки.

Таким образом реализуется упрощение автоматической обработки документа и его отображение в зависимости от окружения. Один и тот же файл может совершенно разным образом отображаться на экранах компьютера, мобильного телефона, в

печати, так как свойства этих устройств вывода существенно различаются.

Однако это правило часто нарушается: например, создавая документ в редакторе наподобие MS Word, пользователь может выделять заголовки жирным шрифтом, но нигде не указывать, что эта строка является заголовком.

Также возможно изолировать разметку от содержания документа, используя указатели, метки, идентификаторы или другие методы координации. Такая «отделенная разметка» характерна для внутреннего представления программ, работающих с размеченными документами. Тем не менее, внедренная или «междустрочная» разметка более принята где бы то ни было.

1.7 Отношение к языкам программирования

Разумеется, что языки гипертекстовой разметки имеют все свойства, присущие языкам как знаковым системам, но их статус как языков программирования часто ставится исследователями и пользователями Всемирной паутины под вопрос.

Рассмотрим определение языков программирования — это совокупность однообразных, определенных и понятных исполнителю инструкций для последовательной их записи с целью исполнения неким устройством, являющимся частью вычислительной машины. Язык программирования также должен обладать следующими особенностями:

- ограниченное число заранее известных инструкций;
- заранее известные правила структуры инструкций;
- каждая конструкция языка должна однозначно соотносить то, что написано, с тем, что требовалось обозначить.

Из вышесказанного следует, что языки гипертекстовой разметки попадают под определение языков программирования, но не являются ими, так как их задача заключается в структурировании и формировании данных, а не обработке.

Также языки гипертекстовой разметки не являются Тьюринг-полными — на них можно совершить только некоторую ограниченную совокупность действий, интерпретируемых средствами вывода.

2. История развития

Даже в контексте данного исследования историю развития языков гипертекстовой разметки нельзя рассматривать самостоятельно, в отрыве от более общих понятий. Поэтому будет уместно рассмотреть истории развития гипертекста и языков разметки отдельно, а затем резюмировать их к объекту исследования.

2.1 История развития гипертекста

Как уже упоминалось ранее, гипертекст как явление появился задолго до Всемирной паутины.

Толкование Книги псалмов Гильберта Порретанского из Пуатье, датируемой, 1150 годом нашей эры, принято считать самой первой системой гипертекста. В другие места книги читателя отсылали специальные пометки на полях. Бумажный носитель, естественно, не позволял пользоваться такой системой с достаточным удобством.

Первая теория автоматизации процесса создания и удобного использования гипертекстовой системы принадлежит Ванневару Бушу, который представлял её в виде машины, оперирующей микрофотопленками. Эта машина, названная им «Мемексом», была описана им в 1945. Разумеется, ни о какой возможности перехода запроса от одного «Мемекса» к другому не было речи — машина была задумана в виде автономного письменного стола.

Сам термин «изобретён» Теодором Нельсоном в 1965 году, а через семь лет он же предложил систему визуализации «параллельных документов» — «Ксанаду», наглядно представлявшую ссылки и связи документов.

Билла Аткинсон был вдохновлен «Ксанаду» на создание программы «Гиперкард» — она служила основой для ряда современных систем гипертекста и определила главные инструменты навигации браузеров.

Множество основных идей и изобретений в области гипертекста так и остались нереализованными или неудачно сформированными в определяющих массовое представление о гипертексте современных браузерах. В целом, как и тысячу лет назад, составить хороший гиперссылочный аппарат всё ещё достаточно затруднительно, и, кроме скорости перемещения по ссылкам, мало что изменилось.

2.2 История развития языков разметки

Понятие «разметка», «markup» произошло от словосочетания «marking up» — «процесс помечивания, размечивания». Оно, в свою очередь, пришло из издательской практики проставлять специальные условные пометки в тексте и на полях рукописей и корректур перед сдачей их в печать.

Таким образом «markup men» — «разметчики» — указывали спецификацию переданной работы: стиль, гарнитуру, размер шрифта и так далее, указанные для каждой конкретной части текста. Сейчас разметкой текста занимаются корректоры, редакторы, графические, а также веб-дизайнеры и сами авторы.

Можно сказать, что переход языков разметки с традиционных бумажных носителей на компьютерную операционную среду начался с понятия «Универсального кодирования» или «generic coding» — так назвал свою идею использовать языки разметки в компьютерной обработке текстов Вильям Танниклифф на конференции в 1967 году. Он руководил разработкой соответствующего стандарта GenCode для издательской индустрии в 1970-е годы и занимал важный пост в одном из комитетов ISO.

Тем не менее, «отцом» языков разметки всё таки называют Чарльза Голдфарба из IBM. Их концепция родилась у него в 1969 году, при работе над очень примитивной системой управления документами, предназначенной для адвокатских контор. В том же году он принял участие в создании языка IBM GML, представленного в 1973 году.

Следующим витком эволюции языков разметки, стало появление стандарта в 1978 году SGML, основанного на GML и GenCode, который был принят организацией ISO в качестве стандартного в октябре 1986 года.

Доступность издательского ПО с функцией WYSIWYG вытеснила большинство этих языков среди обычных пользователей, хотя серьёзная издательская работа по-прежнему использует разметку для специфических не визуальных структур текста, а WYSIWYG-редакторы сейчас чаще всего сохраняют документы в форматах, основанных на языках разметки.

К 1991 году использование SGML было ограничено бизнес-программами и базами данных, а WYSIWYG-инструменты использовались для остальных программ обработки документов. Ситуация изменилась, когда сэр Тим Бернерс-Ли, узнав о

SGML от своего коллеги Андерса Бергланда и других сотрудников в ЦЕРНе, использовал синтаксис SGML для создания HTML. Этот язык имел сходство с другими основанными на синтаксисе SGML языками разметки, однако начать его использование было гораздо проще даже тем разработчикам, которые никогда этого не делали.

XML (расширяемый язык разметки) — это широко используемый в настоящее время язык разметки. Он разработан W3C в возглавляемом Джоном Босаком комитете. Его основное предназначение быть проще SGML и фокусироваться на документах в интернете. Пользователям разрешается создавать любые теги, какие необходимы. Становлению XML помогли — каждый XML-документ может быть написан тем же способом, как и SGML-документ, а программы и пользователи, использующие SGML, могли перейти на XML достаточно легко.

Тем не менее, XML лишился многих ориентированных на людей особенностей языка SGML, упрощавших его использование. Другие улучшения исправляли некоторые проблемы SGML на международном уровне и делали возможным разбор документа иерархически, даже если он не был доступен DTD.

Начиная с января 2000 года, все рекомендации для W3C были основаны на XML в большей степени, чем на SGML, была предложена аббревиатура XHTML. Спецификации языка требовали, чтобы XHTML-документы были оформлены как XML-документы, это позволяет использовать XHTML для более четких и точных документов, используя теги от HTML.

На данный момент основными действующими и разрабатываемыми языками (гипертекстовой) разметки являются HTML, как монополист разметки в веб-среде, и XML, являющийся гибким и удобным инструментом для разработчиков ПО.

Резюмируя этот раздел, можно сказать, что SGML, ставший наследником идей GenCode и GML, а также произошедшие от них HTML и XML, развивались в полной зависимости от операционной среды, в которой действуют их разработчики. Это во многом повлияло не только на их развитие, но и на его прекращение — так можно сказать, например, про GML, специфика которого обуславливала его применение исключительно для программного обеспечения компании IBM.

3. Особенности основных языков гипертекстовой разметки

В этом разделе чуть более подробно рассматриваются основные, или ключевые, языки (гипертекстовой) разметки.

Стоит отметить, что для самых первых языков (гипертекстовой) разметки недостаточно актуальной литературы, а их разработка давно остановлена, поэтому они будут рассмотрены вкратце. Тем не менее, их операционная среда вполне воспроизводима современными средствами, поэтому каждый может опробовать себя в роли пользователя редакторского программного обеспечения конца прошлого века.

3.1 GML

Начать изучение этого раздела стоит именно с GML, или Generalized Markup Language, одного из главных предков существующих языков гипертекстовой разметки. Он является набором макросов, использующих процедурную разметку SCRIPT для формирования разметки документа. SCRIPT, в свою очередь, это основной компонент продукта Document Composition Facility, использующийся для составления документов в IBM — стартовый набор тегов в GML поставлялся вместе с ним.

Основные компоненты GML были также реализованы в издательских системах для мэйнфреймов, да и сам язык пользовался довольно большой популярностью в отрасли.

GML позволяет форматировать и редактировать текст, поддерживает обмен документами подсистемами извлечения информации. При этом он представляет собой концепцию формально определенных типов документов со строгой иерархией структурированных элементов.

При использовании GML документ помечается тегами, которые определяют, чем является текст с точки зрения параграфов, заголовков, списков, таблиц. Такой документ может быть автоматически отформатирован для различных устройств, остается только указать разновидность устройства. Например, можно форматировать документ для лазерного или матричного принтера, или просто вывести на экран, всего лишь указав профиль оборудования, без изменения самого документа.

```
:h1.Chapter 1: Introduction
:p.GML supported hierarchical containers, such as
:ol
:li.Ordered lists (like this one),
:li.Unordered lists, and
:li.Definition lists
:eol.
as well as simple structures.
:p.Markup minimization (later generalized and formalized in SGML),
allowed the end-tags to be omitted for the "h1" and "p" elements.
```

Рисунок 1, пример GML-скрипта

3.2 SGML

За счет расширения GML такими концепциями, как короткие сноски, обработка ссылок и параллельные типы документов, был создан язык SGML, или Standard Generalized Markup Language, стандартный обобщённый язык разметки.

Основной его целью было создания машиночитаемых документов в правительственных и аэрокосмических проектах. В целом, SGML стал активно применяться в издательском бизнесе, к тому же был принят к использованию в Государственном издательстве США и в 1986 году получил статус международного стандарта, но его сложность затруднила его широкое распространение для повседневного использования. Стандартизован ISO в 1986 году.

Основные части документа SGML можно разделить на три части:

- SGML-декларация;
- DTD, Document Type Definition;
- Контент SGML-документа.

```
<DIV1><HEAD>The Structure of a TEI Text</HEAD>

<P>All TEI-conformant texts contain
<LIST TYPE="bullets">

  <ITEM>a <TERM>TEI header</TERM> (marked
  up as a <GI>teiHeader</GI> element) and
  </ITEM>

  <ITEM>the transcription of the text proper
  (marked up as a <GI>text</GI> element).
  </ITEM>

</LIST></P>

</DIV1>
```

Рисунок 2, часть кода SGML-документа

Три характеристики SGML отличали его от прочих языков разметки.

Описательная разметка SGML

С описательной, а не процедурной, разметкой один и тот же документ может быть обработан разнообразными программами, каждая из которых может применять различные инструкции обработки к тем его частям, которые она считает важными. Например, программа анализа содержимого может полностью игнорировать сноски, тогда как программа форматирования может извлекать и собирать их для печати в конце каждой части. Различные виды инструкций обработки могут ассоциироваться с одной и той же частью файла. Например, одна программа может извлекать из документа фамилии людей и географические названия для создания индекса или базы данных, тогда как другая, обрабатывающая тот же самый текст, может печатать фамилии и названия отличающимся шрифтом.

Типы документов SGML

SGML вводит понятие типа документа, и, соответственно, определения типа документа (document type definition, DTD). Документы считаются типизированными, так же, как и другие обрабатываемые компьютерами объекты. Тип документа формально определяется его составными частями и их структурой. Определение, например, отчета может быть таким, что он состоит из заголовка и, возможно, автора, за которыми следует аннотация и последовательность одного или более

абзацев. Любой документ в отсутствие заголовка, в соответствии с этим формальным определением, не будет формально являться отчетом, так же как не будет им являться и последовательность абзацев, за которой следует аннотация, невзирая на то, насколько похож на отчет такой документ с точки зрения читателя-человека.

Поскольку документы относятся к известным типам, можно использовать специальную программу, называемую анализатором (parser), для того, чтобы обработать документ, утверждающий, что он относится к конкретному типу, и проверить, действительно ли все элементы, требуемые для данного типа документов, присутствуют и находятся в правильной последовательности. Что еще более важно, разные документы одного типа могут обрабатываться унифицированным образом. Можно писать более интеллектуальные программы, использующие знания, заключенные в информационной структуре документа.

Независимость данных SGML

Основная цель проектирования SGML была в создании гарантий того, что документ, закодированный согласно его положениям, будет переносимым с одной аппаратной и программной среды в другую без потери информации. Два его свойства, описанных выше, отвечают этому требованию на абстрактном уровне; третье свойство — на уровне строчек байтов (символов), которые составляют документ. SGML предоставляет обобщенный механизм строковой подстановки, то есть, простой машинно-независимый способ указания, что конкретная строка символов в документе в момент обработки документа должна заменяться на некоторую другую строку. Одно очевидное применение этому механизму — обеспечение единой терминологии; другое, и более значительное, -- противодействие известной неспособности разных компьютерных систем понимать наборы символов друг друга, например, одной системе представлять все графические символы, необходимые приложению, путем описательного отображения непередаваемых символов. Строки, определяемые этим механизмом, называются сущностями (entities).

Атрибуты и объекты SGML

В контексте SGML, слово атрибут, подобно другим, имеет строгий технический смысл. Оно используется для описания информации, являющейся в каком-либо смысле описательной для конкретного появления элемента, но не являющейся частью его содержимого. Например, можно добавить атрибут status к экземплярам

некоторых элементов для обозначения степени их достоверности, или добавить атрибут `identifier`, так что можно будет ссылаться на конкретное появление элемента из других мест документа. Атрибуты полезны именно в таких случаях.

Хотя разные элементы могут иметь атрибуты с одинаковыми названиями (например, в схеме TEI каждый элемент определяется имеющим атрибут `id`), эти атрибуты всегда считаются различными и могут иметь различные присваиваемые им 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 до сих пор аспекты SGML все имели отношение к разметке структурных элементов документа. SGML также предоставляет простой и гибкий метод кодирования и наименования произвольных частей действительного содержимого документа переносимым образом. 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`).

3.3 HTML

Тем не менее SGML был малоизвестен до 1990 года, когда Тим Бернерс-Ли, автор идеи World Wide Web, создал Hypertext Markup Language, являющийся подмножеством SGML. Вскоре после этого все виды документов и данных в начале и в конце каждого текстового элемента стали содержать тэги в стиле `<тэг>`.

HTML является стандартизированным язык разметки документов во Всемирной паутине. Большинство веб-страниц содержат описание разметки на языке HTML. Разметка интерпретируется браузерами, а полученный в результате текст отображается на устройстве вывода. До своей 5-ой версии HTML определялся как приложение SGML.

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 иногда 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 оформлении.

HTML-документы, имеющие расширение `.html` или `.htm`, интерпретируются специальными приложениями, называемые «браузерами» или «интернет-обозревателями», которые отображают документ в его форматированном виде. Такие приложения обычно предоставляют пользователю удобный интерфейс для запроса веб-страниц, их просмотра и/или вывода на иные внешние устройства. Также, при необходимости, могут отправлять введённые пользователем данные на сервер. Сейчас наиболее популярными браузерами являются Google Chrome, Mozilla Firefox, Opera, Яндекс.Браузер и Safari.

Любой HTML-документ представляет собой набор элементов, причём начало и конец каждого элемента обозначается специальными пометками — тегам. Элементы могут быть пустыми, то есть не содержащими никакого текста и других данных. В этом случае обычно не указывается закрывающий тег (например, тег переноса строки `
` - одиночный и закрывать его не нужно). Кроме того, элементы могут иметь атрибуты, определяющие какие-либо их свойства (например, атрибут `href="` у ссылки). Атрибуты указываются в открывающем теге.

Регистр, в котором набрано имя элемента и имена атрибутов, в HTML значения не имеет. Элементы могут быть вложенными.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>HTML Document</title>
  </head>
  <body>
    <p>
      <b>
        Этот текст будет полужирным, <i>а этот — ещё и курсивным</i>.
      </b>
    </p>
  </body>
</html>
```

Рисунок 3, часть кода на языке HTML

Кроме элементов, в HTML-документах есть и сущности — «спецсимволы». Сущности начинаются с символа амперсанда и имеют вид `&имя;` или `&#NNNN;`, где `NNNN` — код символа в Юникоде в десятичной системе счисления.

Например, `©` — знак авторского права (©). Как правило, сущности используются для представления символов, отсутствующих в кодировке документа, или же для представления «специальных» символов: `&` — амперсанда (&), `<` — символа «меньше» (<) и `>` — символа «больше» (>), которые некорректно записывать «обычным» образом, из-за их особого значения в HTML.

Хронология версий HTML

- HTML 1.0 — официально не существует;
- HTML 2.0 — 24 ноября 1995 года;
- HTML 3.0 — 28 марта 1995 года — IETF Internet Draft (до 28 сентября 1995 года);
- HTML 3.2 — 14 января 1997 года;
- HTML 4.0 — 18 декабря 1997 года;
- HTML 4.01 — 24 декабря 1999 года;
- IISO HTML, основан на HTML 4.01 Strict — 15 мая 2000 года;
- HTML5 — 28 октября 2014 года;
- HTML 5.1 — 1 ноября 2016 года;
- HTML 5.2 — 14 декабря 2017 года.

3.4 XML

Можно сказать, что XML — Extensible Markup Language, то есть Расширяемый Язык Разметки, — возник в результате развития языка HTML. Однако было бы грубой ошибкой воспринимать его лишь как некую усовершенствованную версию языка HTML. По сути, язык XML представляет собой новое поколение языков разметки.

С физической точки зрения XML-документ состоит из сущностей, из которых каждая может отсылать на другую сущность. Единственный корневой элемент — документная сущность. Содержание сущностей — символы.

С логической точки зрения документ состоит из комментариев, объявлений, элементов, ссылок на сущности и инструкций обработки. Всё это в документе структурируется разметкой.

Сущность — мельчайшая часть в документе. Все сущности имеют некое содержание, и у всех них есть имя. Проще говоря, термин «сущность» описывает «сущую вещь», «что-то».

Документ состоит из сущностей, содержание которых — символы. Все они разделены на два типа: символьные данные и разметки. К разметке принадлежат теги, обозначающие границы элементов, объявления и инструкции обработки, включая их атрибуты, ссылки на сущности, комментарии, а также последовательности символов, обрамляющие секции «CDATA». Часть документа, не принадлежащая разметке, составляет символьные данные документа.

Все составляющие части документа обобщаются в пролог и корневой элемент. Корневой элемент — обязательная часть документа, составляющая всю его суть (пролог, вообще говоря, может отсутствовать). Может включать (а может не включать) вложенные в него элементы и символьные данные, а также комментарии. Вложенные в корневой элемент элементы, в свою очередь, могут включать вложенные в них элементы, символьные данные и комментарии, и так далее. Пролог может включать объявления, инструкции обработки, комментарии. Его следует начинать с объявления XML, хотя в определённой ситуации допускается отсутствие этого объявления.

Элементы документа должны быть правильно вложены: любой элемент, начинающийся внутри другого элемента (то есть любой элемент документа, кроме

корневого), должен заканчиваться внутри элемента, в котором он начался. Символьные данные могут встречаться внутри элементов как непосредственно, так и в специальных секциях «CDATA». Объявления, инструкции обработки и элементы могут иметь связанные с ними атрибуты. Атрибуты используются для связывания с логической единицей текста пар имя-значение.

Разметка всегда начинается символом < и заканчивается символом >. Наряду с символами < и >, специальную роль для разметки играет также символ &. Угловые скобки обозначают границы элементов, инструкций обработки и некоторых других последовательностей. Амперсанд позволяет выполнить замену текста при помощи *сущностей*.

В языке XML все имена должны начинаться с буквы, символа подчёркивания (_) или двоеточия (:) и продолжаться только допустимыми для имён символами, а именно они могут содержать только буквы, входящие в секцию букв кодировки Unicode, арабские цифры, дефисы, знаки подчёркивания, точки и двоеточия. Однако имена не могут начинаться со строки xml в любом регистре. Имена, начинающиеся с этих символов, зарезервированы для использования консорциумом W3C. Нужно помнить, что, так как буквы не ограничены исключительно символами ASCII, то в именах можно использовать слова из родного языка.

Объявление XML указывает версию языка, на которой написан документ. Поскольку интерпретация содержимого документа зависит от версии языка, то Спецификация предписывает начинать документ с объявления XML. В первой (1.0) версии языка использование объявления не было обязательным, в последующих версиях оно обязательно. Таким образом, версия языка определяется из объявления, и если объявление отсутствует, то принимается версия 1.0.

Для объявления типа документа существует специальная инструкция !DOCTYPE. Она позволяет задать при помощи языка DTD, какие в документ входят элементы, каковы их атрибуты, какие сущности могут использоваться и другое.

Инструкции обработки позволяют размещать в документе инструкции для приложений. В следующем примере показана инструкция обработки, передающая xml-stylesheet-приложению (например, браузеру) инструкции в файле my-style.css посредством атрибута href: <?xml-stylesheet type="text/css" href="mystyle.css" ?>

Комментарии не относятся к символьным данным документа. Комментарий начинается последовательностью «<!--» и заканчивается последовательностью «-->», внутри не может встречаться комбинация символов «--». Символ & не

используется внутри комментария в качестве разметки.

Корректный документ соответствует всем общим правилам синтаксиса XML, применимым к любому XML-документу: правильная структура документа, совпадение имен в начальном и конечном теге элемента и т. п. Документ, который неправильно построен, не может считаться документом XML.

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE recipe>
<recipe name="хлеб" preptime="5min" cooktime="180min">
  <title>
    Простой хлеб
  </title>
  <composition>
    <ingredient amount="3" unit="стакан">Мука</ingredient>
    <ingredient amount="0.25" unit="грамм">Дрожжи</ingredient>
    <ingredient amount="1.5" unit="стакан">Тёплая вода</ingredient>
  </composition>
  <instructions>
    <step>
      Смешать все ингредиенты и тщательно замесить.
    </step>
    <step>
      Закрыть тканью и оставить на один час в тёплом помещении.
    </step>
    <!--
      <step>
        Почитать вчерашнюю газету.
      </step>
      - это сомнительный шаг...
    -->
    <step>
      Замесить ещё раз, положить на противень и поставить в духовку.
    </step>
  </instructions>
</recipe>
```

Рисунок 4, пример кулинарного рецепта, размеченного с помощью XML

Пока развивалась идея использовать языки разметки с текстовыми документами, увеличилось использование языков разметки в других областях, предложили использовать их для представления различных типов информации, включая плей-листы, векторную графику, веб-сервисы, интерфейсы пользователя. Большинство этих приложений основано на XML, так как это хорошо структурированный и расширяемый язык.

Заключение

В рамках данной курсовой работы были рассмотрены языки гипертекстовой разметки, которые особенно актуальны в современной цифровой среде. Было определено понятие «Языки гипертекстовой разметки», а также выявлены их основные особенности.

Помимо этого, была изучена их история и проанализированы особенности конкретных языков (гипертекстовой) разметки, а именно — GML, SGML, HTML и XML.

Библиография

1. *Лебедев А.* Ководство. — 5-е изд. — М.: Издательство Студии Артемия Лебедева, 2018. — 556 с.
2. *Lambert L.* The Internet: Biographies. — 2005. — 128 с.
3. *Хантер Д., Рафтер Д., Фаусетт Д., Влист Э., и др.* XML. Работа с XML, 4-е издание — М.: «Диалектика», 2009. — 1344 с.
4. *Шафер С.* HTML, XHTML и CSS. Библия пользователя, 5-е издание — М.: «Диалектика», 2010. — 656 с.
5. *Фримен Э., Фримен Э.* Изучаем HTML, XHTML и CSS — П.: «Питер», 2010. — 656 с.
6. *Сухов К.* HTML5 — путеводитель по технологии. - 2 изд. - М.: ДМК Пресс, 2013.
7. *Босуэлл Д., Фаучер Т.* Читаемый код, или Программирование как искусство. — СПб.: Питер, 2012. — 208 с.
8. *Ташков П. А.* Веб-мастеринг. HTML, CSS, JavaScript, PHP, CMS, AJAX, раскрытка — СПб.: Питер, 2010 — 512 с.
9. SGML (Standard Generalized Markup Language) // Национальная библиотека им. Н. Э. Баумана URL: [https://ru.bmstu.wiki/SGML_\(Standard_Generalized_Markup_Language\)](https://ru.bmstu.wiki/SGML_(Standard_Generalized_Markup_Language)).
10. HTML (HyperText Markup Language) // Национальная библиотека им. Н. Э. Баумана URL: [https://ru.bmstu.wiki/HTML_\(HyperText_Markup_Language\)](https://ru.bmstu.wiki/HTML_(HyperText_Markup_Language))
11. XML (eXtensible Markup Language) // Национальная библиотека им. Н. Э. Баумана URL: [https://ru.bmstu.wiki/XML_\(eXtensible_Markup_Language\)](https://ru.bmstu.wiki/XML_(eXtensible_Markup_Language))