Руслан Богатырев

Колесо обозрения

Источник: Мир ПК, #10/1997.

Аналитические заметки по ключевым публикациям. Лето 1997 г.: HTML и XML

Детские воспоминания

Кто из нас в детстве не любил аттракционы? Всевозможные карусели, комнаты смеха, горки, лодочки, качели, автопоезда. Все это здорово. И все же... все же посередине шумного и веселого парка развлечений особняком стоит нечто большое и загадочное, чему взрослые дали такое странное название - чертово колесо. Куплены билеты, и вот, расположившись в уютных качающихся люльках, мы с замиранием сердца смотрим вниз - туда, где нашему взору неспешно и неожиданно открывается захватывающее зрелище. Казалось, еще минуту назад мы стояли на земле, и ничто не могло нас удивить. А теперь, подобно птице, мы парим в воздухе и с нескрываемым интересом разглядываем раскинувшиеся внизу окрестности.

Прошли годы, и мы с головой окунулись в безумный водоворот слухов, событий, фактов, из которых с превеликим трудом удается извлечь хотя бы малую толику полезной информации. Ну а если дело касается компьютеров и новых информационных технологий, то тут уж без знания подводных камней и всех хитросплетений ощущаешь себя просто потерянным человеком - чужестранцем, которого судьба забросила в далекую страну со знакомым языком, но непонятной культурой. Листая прессу и путешествуя по Internet, то и дело ловишь себя на мысли, что взгляд привычно и быстро, почти не останавливаясь, скользит по строчкам новостей и лишь изредка задерживается на чем-то понятном или сенсационном.

И все же попробуем разорвать заколдованный круг, замедлить нескончаемый бег и, следуя принципу знакомого с детства аттракциона, вглянуть на вроде бы известные нам вещи и факты с разной высоты - быть может, тогда нам удастся лучше понять и почувствовать их суть?

Летние месяцы обычно небогаты событиями, зато на поверхности ненадолго успокоившегося океана заметны любые, даже самые незначительные изменения. Сезон отпусков принес с собой немало интересного, но я бы выделил две публикации. Они различны по своему характеру и направленности: первая представляет собой статью, напечатанную в ориентированном на широкую аудиторию американском журнале *Internet World,* вторая - краткое техническое исследование, проведенное фирмой Microsoft и помещенное на ее сервер. Обе публикации в той или иной степени предчувствовали и предвосхищали ключевое событие нынешнего лета - "тихий переворот", который, выпустив в июле спецификации HTML 4.0 и XML, осуществил консорциум W3C (The World Wide Web Consortium). Но обо всем по порядку. Давайте поудобнее устроимся в креслах и совершим небольшое путешествие на нашем колесе обозрения.

Еще больше мощи у нашего браузера?

В июньском выпуске журнала *Internet World* была опубликована статья Аарона Вайса под броским заголовком "Динамический HTML: *еще больше мощи у вашего браузера?" (Aaron Weiss. Dynamic HTML: More Power in Your Browser? //* Internet World, 1997, Vol.8, N 6). Как известно, фирмы Microsoft и Netscape Communications в рамках дальнейшего развития языка HTML недавно предложили две концепции, объединенные названием Dynamic HTML. Суть их вкратце состояла в том, чтобы изменить статическую природу доминирующих ныне в Internet HTML-страниц, добавив возможность динамически управлять цветовым и шрифтовым оформлением, позиционировать на странице различные элементы, воспроизводить визуальные эффекты. Основы для реализации этих решений были заложены опубликованным еще в декабре прошлого года проектом CSS1 (Cascading Style Sheets), регламентирующим правила формирования визуального контекста, в котором отображается

содержимое не только конкретной страницы, но и других страниц с данного Web-сервера (в частности, за счет подключения общего стилевого файла командой <LINK>).

Вайс обращает внимание читателя на то, что с приходом динамического HTML не просто меняется внешний облик HTML-страниц, суть гораздо глубже - продолжается постепенный перенос программной обработки с Web-сервера на клиентскую машину пользователя. Так и хочется сказать, что малыши-браузеры начинают потихоньку набираться ума-разума и уже не бегают к родителям-серверам за каждой малостью, а сами в своей песочнице решают многие жизненные проблемы. Несмотря на то что такие языки, как JavaScript/JScript и VBScript, смогли создать прикладной слой, исполняемый на локальном компьютере, они в силу отсутствия единства среди производителей браузеров так и не смогли решить задачу унификации процедурных аспектов Web-страниц. И все же, с точки зрения Вайса, сам по себе динамический HTML - важный шаг вперед. Он не только обеспечивает локальное изменение "на лету" самого Web-документа, но и предоставляет разработчикам более широкие возможности по управлению внешним обликом и реакцией страницы. За счет усиления обратной связи с пользователем Web-страница становится более *интерактивной.*

И Netscape, и Microsoft, пишет Вайс, с энтузиазмом воплощают идею динамического HTML, но вот подходы к ее реализации у фирм разные. Главный пункт разногласий - позиционирование элементов на странице. Microsoft предпочитает не трогать синтаксис HTML, добиваясь изменения функциональности за счет расширения параметров стилевых таблиц. Netscape же, считая это искусственным усложнением, вводит новые команды (теги) с именами LAYER и ILAYER. При этом реализация динамики отображения у Netscape ложится на JavaScript-подобные стилевые таблицы, а у Microsoft - на JScript, VBScript и ActiveX, что ведет к несовместимости. "Возможен ли в такой войне победитель?" - вопрошает в заключение Вайс.

Динамический HTML: мнение Microsoft

Пытаясь развеять какие бы то ни было сомнения относительно исхода конфликта, корпорация Microsoft опубликовала на своем Web-сервере краткий сравнительный анализ обоих подходов (см. http://www.microsoft.com/workshop/author/dhtml/?), подчеркнув, что реализация Netscape обладает тремя нестандартными механизмами: это специальные стилевые таблицы (JASS, JavaScript Accessible Style Sheets), множественные визуальные уровни (layers) и динамические шрифты. Даже поверхностное знакомство с этим документом убеждает в недостаточной искренности его анонимных создателей. Раз уж Microsoft поместила свой анализ в Internet и постоянно его модифицирует, я для корректности укажу его дату - 15 июля. Будет время - обязательно изучите: там есть немало занимательного, начиная от сравнительных колонок, в которых в графе Microsoft нет ни одного (!) прочерка относительно поддержки каких бы то ни было возможностей (а в графе Netscape - вагон и маленькая тележка), и кончая постоянными упреками в адрес Netscape относительно ограниченной реализации объектной модели документа (DOM, Document Object Model). Неужели и впрямь все так печально?

В чем же выражается эта ограниченность? Оказывается, в трех моментах: в жесткой привязке к JavaScript (ненароком подумаешь, что Microsoft и в самом деле обеспечивает языковую независимость и полнокровно поддерживает тот же JavaScript), в избирательности генерирования событий (у Netscape этой способностью обладают не все объекты) и в невозможности выхода событий за пределы документа (event bubbling).

Самое интересное, что здесь, пользуясь заведомой неосведомленностью читателя (кто же будет сидеть и сверять нюансы множества нормативных документов?), неизвестный автор из Microsoft явно лукавит. Дело в том, что объектная модель документа появилась "задним числом", да и то с единственной целью внести определенность и хоть как-то объединить на общей основе HTML, CSS и XML (о нем речь пойдет ниже). В начале июля W3C представил на всеобщее обозрение проект спецификации DOM, в котором выделяются три уровня. Нулевой закрепляет де-юре то, что уже существует де-факто (W3C подчеркивает, что за точку отсчета берутся те возможности, которые уже имеются в Microsoft Internet Explorer 3.0 и Netscape Navigator 3.0). Наметки первого уровня были представлены 11 июля, а его выпуск ожидался в конце августа, но затем был перенесен на конец сентября. Сроки выхода второго даже не названы. В то же время в проекте спецификации DOM предварительно оговариваются те моменты, которые планируется предусмотреть в рамках второго уровня. Среди них - событийная модель (Event Model), которая и должна обеспечивать "поголовную" генерацию событий всеми объектами, а также последующее распространение этих самых событий.

Какие тут могут быть комментарии? Все это лишний раз убеждает в том, что к документам, подобным анализу Microsoft, надо относиться с большой осторожностью.

Консорциум W3C водружает знамя XML

Легкий ветерок приятно дует в лицо, облака понемногу заволакивают небо, но по-прежнему убаюкивающе поскрипывают крепления соседних кабинок нашего чертова колеса. Еще не успела обсохнуть краска на свежеотпечатанном номере *Internet World,* как 8 июля консорциум W3C объявил о выпуске первого рабочего проекта HTML 4.0. И в тот же день знаменитый Тим Бернерс-Ли, директор W3C и идеолог Всемирной паутины, выступил с обращением, в котором изложил свое видение перспектив нового поколения HTML (см. http://www.w3.org/pub/ WWW/TR/). Похвалив продуктивность рабочей группы по HTML, куда вошли представители таких фирм, как Adobe Systems, Hewlett-Packard, IBM, Microsoft, Netscape Communications, Novell и Sun Microsystems, он подчеркнул, что HTML 4.0 вобрал в себя не только новые средства поддержки гипертекста и мультимедиа, но и механизм гибкого расширения функциональных возможностей HTML, в частности, за счет нового языка XML (Extensible Markup Language - расширяемый язык разметки).

Язык XML (см. Extensible Markup Language // W3C Working Draft, 07 Aug 1997) с учетом активной поддержки W3C ныне преподносят как панацею, способную вылечить всерьез заболевшую Сеть. Его рассматривают как упрощенное и творчески переработанное с учетом нужд Internet подмножество SGML (первоначально XML называли WebSGML). Строго говоря, XML, как и SGML, - это не язык в обычном понимании этого слова, а метаязык, т. е. средство, определяющее правила своего расширения другими языками. Работая на нем, вы можете описывать собственные теги, а следовательно, операторы своего языка и их параметры.

При рассмотрении SGML, HTML, XML и других языков этого семейства важно правильно понимать, что такое метаязык. Если толковать этот термин в лоб, на уровне здравого смысла, то можно прийти к выводу, что коль скоро SGML является языком для описания языков (что верно), то документ (программа) на XML не является документом на SGML, а это уже неверно. Несколько упрощая проблему, можно сказать, что SGML - в принципе такой же полноправный язык, как и XML; просто его грамматика задает обобщенное пространство, в котором можно найти место другим языкам, ужесточающим (конкретизирующим) требования. Другими словами, SGML - это общий случай, XML - частный, а HTML - еще более частный (но стоящий особняком по отношению к XML). Задумывались ли вы над тем, что библиотека процедур для языка программирования определяет иной язык, конкретизирующий исходный? Другой пример - средства управления компилятором (прагмы), активно используемые в ряде реализаций разных языков внутри блока комментария. Паразитируя на известных исходному языку структурах (комментариях), разработчики, по сути, задают свой собственный, конкретизирующий язык. Как видите, все довольно просто и чем-то напоминает матрешку.

Язык XML, подобно SGML, позволяет определять правила описания документов самых разных типов. При этом чересчур ограниченный язык стилей CSS заменяется на оформленный в рамках XML язык XSL (Extensible Style Language), который, в свою очередь, опирается на более гибкий язык DSSSL (Document Style Semantics and Specification Language; см. http://www.jclark.com/dsssl; в 1996 г. для него был принят международный стандарт ISO-10179). Важную роль в развитии DSSSL играет язык Scheme (диалект Лиспа). Стоит отметить, что многие заслуживающие внимания идеи по структуре и топологии гиперсвязей XML позаимствованы из языка HyTime, стандартизированного (ISO-10744) при активном участии автора SGML Чарльза Гольдфарба, и из проекта Text Encoding Initiative (TEI P3).

XML: что он нам несет?

Этот раздел адресован в первую очередь тем, кто восхищается любыми нововведениями и, полагаясь на мудрость W3C, готов стать апологетом XML и отдать жизнь за новую веру. Прочитайте эти строки и призадумайтесь, стоит ли так уж безусловно вверять свою судьбу новоявленным жрецам компьютерных наук?

Впервые проект XML, созданный всего за несколько месяцев специальной коллегией SGML Editorial Review Board (SGML ERB) при поддержке W3C, был представлен в ноябре 1996 г. на конференции SGML-96 в Бостоне. Идейные основы XML и обоснование его необходимости изложил в своей статье Джон Босак из компании Sun Microsystems - руководитель SGML ERB и, пожалуй, главный идеолог XML (см. http://sunsite.unc.edu/pub/sun-info/standards/xml/why/xmlapps.htm).

Стратегия W3C (не особенно, правда, афишируемая) состоит в том, чтобы упредить момент, когда дамба HTML окончательно прорвется и откроет дорогу бушующей стихии, способной полностью затопить Internet. W3C намеревается соорудить взамен прежней дамбы новую с именем XML, не разрушая при этом отслужившие свой срок старые постройки.

Разные языки семейства XML начинают расти как грибы после дождя. 10 июля появился новый проект спецификации языка MathML - приложения XML для описания математических формул. Другими направлениями, уже оформленными в виде прикладных языков, являются CML (Chemical Markup Language) - язык описания химических формул, HCML (Health Care Markup Language) - язык описания информационных связей в области здравоохранения и OFX (Open Financial Exchange) - язык обмена финансовой информацией. Особо следует выделить OSD (Open Software Description) - язык описания дистрибутивной и рабочей структуры программных продуктов, позволяющий обеспечить основу для автоматического обновления (через Internet) установленного ПО. Назову также Microsoft CDF (Channel Definition Format) - XML-приложение для реализации различных операций в рамках push-технологии. Для программного взаимодействия с XML Microsoft предлагает использовать специальный API-интерфейс (XAPI).

Но кто же будет обрабатывать многочисленные XML-языки? Судя по всему, соответствующие языковые процессоры (XML-процессоры), встроенные в браузер или реализованные как подключаемые модули (plug-ins). Естественно спросить, чем же это лучше обычного механизма работы со встроенными объектами на уровне подключаемых модулей? Только тем, что анархичный процесс регулируется общей синтаксической XML-архитектурой? Сомнительное преимущество. Языков будет не два и не три, а о стандартизации нескольких десятков языков договориться вряд ли проще, чем о стандартизации одного. Похоже, что с введением XML всех начнут причесывать под одну гребенку. В прокрустово ложе весьма своеобразного синтаксиса XML будут запихивать все новые и новые языки.

Существуют два мифа, связанных с XML (см. Design Principles for XML // Draft DD-96-0001, 25 Aug 1996). Миф первый: "XML-документы будет проще создавать вручную". Но откуда такая уверенность? Если имеется возможность определять любую собственную грамматику, то кто гарантирует, что она не окажется еще более запутанной, чем грамматика нынешнего HTML? Уже сейчас многие предлагают отменить из-за их избыточности закрывающие теги XML. Без них куда удобнее, но тогда нарушится неприкосновенное - стандартный синтаксис XML. Миф второй: "Станет легче писать программы, которые будут обрабатывать XML-документы". Написание синтаксических анализаторов действительно упростится, но ведь в конечном итоге это даже не полдела: нужны и все остальные части традиционного компилятора (анализатор промежуточного представления, генератор и оптимизатор кода), и, самое главное, интерпретатор нашего кода. Так что здесь нет никакого выигрыша по сравнению с реализацией встраиваемых объектов с помощью подключаемых модулей, как это делается сейчас с PDF, VRML и Juice.

Но основная проблема XML в том, что "приведение к общему знаменателю" для новоявленного эсперанто требует разбора и интерпретации конкретного XML-языка для данного типа документов (DTD) на стороне клиентской машины. Иначе говоря, по Сети будет передаваться не оттранслированный и оптимизированный компактный код, а текстовые файлы (пусть даже и упакованные), которые на клиентской машине будут обрабатываться XML-процессорами подобно интерпретируемым программам на Бейсике. Результат - потеря производительности и весьма сомнительная гибкость.

Единственными серьезными достижениями рабочей группы по созданию XML являются, на мой взгляд, создание языка метаконтента (MetaContent Language, MCL) и введение куда более богатых средств описания гиперссылок (см. Extensible Markup Language (XML): Part 2. Linking // W3C Working Draft, 31 Jul 1997). MCL позволяет сделать чрезвычайно важную вещь - упорядочить описание отношений внутри содержательной информации. Интересно, что сам язык был в значительной части позаимствован из разработок MCF фирмы Apple, а в основу MCL были положены ориентированные помеченные графы (как тут не вспомнить Дональда Кнута, предвосхитившего метаконтент, но только для программирования). Активное влияние на MCL оказывают MCF фирмы Netscape и XML-Data фирмы Microsoft. Дальнейшее развитие идей MCL в плане унификации обмена метаданными, запросов пользователей и поисковых машин, а также уже практически подготовленный переход на мультиканальную и многоствольную (multicast) Web-архитектуру (концепция WebCanal французского института INRIA), - все это вместе взятое привело к появлению в рамках семейства XML нового языка RDF (Resource Description Framework), который пришел на смену W3C-спецификации PICS-1.1.

В плане описания гиперссылок XML тоже заметен прогресс: появились роли, множественность связей, возможность задать поведение при навигации. Важным аргументом в поддержку XML является то, что в нем учтен опыт готовых проработок по SGML, DSSSL и HyTime. Здесь, в частности, можно отметить разработанный в японских исследовательских лабораториях Fujitsu Labs браузер HyBrick - один из первых SGML-навигаторов текста, в полной мере использующих поддержку интерактивности электронных документов с помощью DSSSL. Кроме того, вполне вероятно, что взамен HTTP-серверов уже в самом ближайшем времени начнут появляться разновидности HEP-серверов (HyTime Engine Peer-to-Peer Protocol), работающие на основе HyTime. Но, как показал симпозиум "W3C Workshop on Push Technology", прошедший 8-9 сентября в Бостоне (США), самый серьезный прорыв готовится с переориентацией неудовлетворительных "подъездных" Web-путей на технологии, предлагаемые фирмами Marimba и @Home Network. Речь идет прежде всего о замене протоколов семейства HTTP на DRP (Distribution and Replication Protocol), который в значительной мере опирается на давно применяемый фирмой Lotus принцип тиражирования данных за счет обмена разностными файлами. Для этой цели подготовлен бинарный формат GDIFF (Generic DIFF), позволяющий обеспечить передачу через Internet только измененной порции информации.

Язык XML, в отличие от HTML, явно ориентируется на улучшение автоматического анализа и извлечения информации с разметкой; удобство работы "писателей" мало кого интересует. А ведь язык, как известно, определяет характер мышления человека. Положение можно сравнить со следующей гипотетической ситуацией. Представьте себе, что кто-то предлагает: давайте не будем больше мучиться с алфавитами, синтаксисом, пунктуацией разных национальных языков, а станем писать только латинскими буквами (обязательно слева направо!) и строить предложения по правилам упрощенного английского языка (basic English). Неужели у вас есть какие-то серьезные аргументы против этого "замечательного" подхода?

История развития компьютерных языков (не только языков программирования) показывает, что долгая и счастливая жизнь языка определяется его здоровьем: он должен пройти испытание временем, доказать свою жизнеспособность на реализации конкретных проектов и лишь по истечении некоторого "инкубационного периода" быть представлен широкой общественности. В случае с XML почти все эти условия были нарушены. На что же тогда надеются те, кто стремится как можно скорее дать ему путевку в жизнь?

Изучая труды создателей идеологии и языков XML, трудно понять, задумывались ли они о том, что станет в ближайшие год-два с совместимостью и производительностью - двумя важнейшими факторами, влияющими на развитие компьютерной индустрии. Удастся ли не выпустить из-под контроля лавинообразный рост XML-языков? Что станет с поисковыми системами? Удастся ли им находить на Web-страницах содержательную информацию, когда она будет с головой укутана в одеяла условных переходов? И как индексировать страницы, превратившиеся по сути в мультимедиа-сцены, где появление и исчезновение тех или иных фрагментов текста зависит от множества разных факторов, и прежде всего от действий пользователя. Большое количество текстовых языков и форматов представления станет все сильнее и сильнее давить своим весом на браузер, "обвешанный" специализированными XML-процессорами. При этом не столько Web-сервер будет разгружаться, сколько каналы связи - засоряться избыточной информацией. И здесь не помогут такие полумеры, как переход с форматов GIF89a и JPEG JFIF на PNG и JPEG/SPIFF, а также переориентация с HTTP на DRP.

Вся беда, на мой взгляд, в том, что исполняемые Web-документы по-прежнему рассматриваются исключительно как документы, а не как программы, а потому есть очень большое подозрение, что скоро все мы станем свидетелями воплощения принципа Чебурашки. Помните, из чувства благородства он предложил усталому крокодилу, доверху нагруженному чемоданами, замечательный выход: "Ген, давай я понесу чемоданы, а ты понесешь меня"? По всей видимости, Web-серверу и промежуточным коммуникационным узлам придется вкалывать и за себя, и за "интеллектуального" клиента.

Так где же выход? Можно создавать базовый "двухэтажный" язык программирования. При этом он должен обеспечивать свое расширение на обоих этажах (системном и прикладном), пройти апробацию и оставаться стабильным хотя бы на протяжении двух-трех лет, а кроме того, позволять четко разграничивать разные по своему назначению языковые слои (подъязыки), объединяющие информацию содержательную (с контекстом национального языка и кодировки), типографскую (шрифтовое оформление), композиционную (расположение элементов), интерфейсную (органы управления), навигационную (описание гиперсвязей), программную (расширения, поведение, реакция объектов), оптимизационную (порядок и условия загрузки и отображения элементов страницы), представительную (аудио-, видео- и печатная формы представления документа), коммуникационную (каналы обмена информацией). В этом случае

можно гарантировать целостную проверку корректности синтаксиса и семантики языка. Можно идти и другим путем: вместо разработки принципиально нового языка сформировать семейство специализированных языков, ориентированных на разные типы документов, и выработать связующий каркас, задающий условия взаимодействия и сосуществования языков с разной ориентацией. Консорциум W3C, активно развивая метаязык XML, сделал ставку на последний подход. Что ж, время покажет, насколько правильным и разумным оказался этот выбор.

Полуторная Паутина

Мерно покачиваясь в уютных люльках, мы приближаемся к концу нашей экскурсии. Глядя на тот азарт и спешку, с которой, не разгибая спины, трудятся разработчики XML, поневоле задаешься вопросом: а можно ли было заранее предусмотреть тот тупик, в который забрел HTML? Ответить на него читатель сможет и сам. Я же лишь подчеркну, что на фоне лихорадочных попыток реанимировать увядающий на глазах HTML гораздо более привлекательной выглядит инициатива фирмы Silicon Graphics, которой дано звучное имя Second Web - "вторая Паутина" (см. Эдвард Маккракен. Moving the World to the Second Web // Мультимедиа. Цифровое видео, 1997, # 5/6, с.54) и в которой центральная роль отводится концепции трехмерных виртуальных миров и соответствующим языкам описания и навигации в этом пространстве, а также поведению пассивных и активных персонажей - "воплощений" (avatars). Среди языков и стандартов, задействованных в Second Web, - VRML, ActiveVRML, D/D96, Living Worlds. В отличие от HTML ("первой Паутины"), здесь, по крайней мере, понятно, чего хотят идеологи этого движения. Правда, работать во "второй Паутине" при нынешних скоростях не просто утомительно, а подчас невыносимо. Но даже если отвлечься от таких "мелочей", ясно, что "вторая Паутина" не заменит первую, точно так же, как и увлечение цветной фотографией и живописью с применением самой изощренной техники исполнения никогда не заменит карикатуру и графику. Тем, кто не собирается во "вторую Паутину" и не в силах ждать, чем же кончится эпопея с XML, остается лишь создавать свою доморощенную "полуторную Паутину", надеясь со временем соединить ее прочными нитями с другими, то и дело разрушаемыми порывами ветра шаткими строениями большой планеты под названием Internet.

Мы уже коснулись земли и почти закончили свое путешествие на колесе обозрения. Надеюсь, что даже после одной прогулки вы посмотрите на окружающие вас знакомые предметы совсем другими глазами. Как гласит древняя истина, новое - это хорошо забытое старое; вопрос лишь в том, смогут ли потомки хотя бы приблизиться к тем идеям, которые когда-то заложили их мудрые предки.

В заключение не могу не поблагодарить А.Ю. Хлесткова за его ценные критические замечания, а также М.С. Суханову за ее большую и кропотливую работу по подготовке статьи к печати.

HTML 4.0 и динамический HTML

Разные обозреватели и специалисты оценивают сложившуюся ситуацию как состояние войны между Microsoft и Netscape. Они, конечно, сгущают краски, но в любом случае положение вещей не может не удручать, особенно если вдуматься в возможные последствия противостояния. Во-первых, каждый, кто познакомится с новыми средствами обоих вариантов динамического HTML, встанет перед дилеммой - либо оставить все как есть, либо соблазниться заманчивыми возможностями. В последнем случае, скорее всего, придется делать нелепый выбор: с кем ты, разработчик, - с Microsoft или с Netscape? Ведь сопровождать два несовместимых варианта страниц с активным использованием изощренных алгоритмов отображения для многих окажется почти невозможным (а главное, ради чего?). Во-вторых, многие пользователи Internet все чаще будут видеть на экране непрогнозируемую реакцию - неподдельное удивление своего любимого браузера, который, обрабатывая очередную страницу, беспомощно застынет. В результате резко затормозится едва-едва наметившийся процесс "облагораживания" Internet-общения.

Немудрено, что первый вопрос, который невольно возникает в связи с HTML 4.0, - это как он соотносится с динамическим HTML, т. е. на чьей стороне W3C - Microsoft или Netscape? Если внимательно вчитаться в текст проекта W3C, то выясняется, что от динамического HTML (что в интерпретации Microsoft, что в интерпретации Netscape) его отделяет очень приличная дистанция. Формально W3C остался "над схваткой"; его нейтралитет фиксируется в документе под названием Document Object Model, где недвусмысленно заявлено, что ""динамический HTML" - это всего лишь термин, используемый разными производителями ПО для описания такого сочетания HTML, стилевых таблиц и программных вставок (scripts), которое добавляет в документы анимацию". Идя по пути компромиссов, W3C занимает гибкую, даже чересчур гибкую позицию - проект HTML 4.0 не отрицает использования возможностей динамического HTML, при

этом фактически устраняясь от конкретизации декларируемых свойств и механизмов. Выясняется, что основа динамики - иерархические стилевые таблицы - совсем не обязаны удовлетворять спецификации CSS1. Более того, вовсе не обязателен и механизм иерархии стилей (cascading), определяющий вложенность контекстов описания разных атрибутов. Причем речь идет не о той или иной стилевой таблице, а о языке описания таких таблиц! Интересно, а как же HTML 4.0 решает споры вокруг проблемы позиционирования элементов (абсолютного, относительного, послойного)? В основном и это отдается на откуп конкретным реализациям.

Давным-давно назревшая проблема динамической транспортировки отсутствующих шрифтов также долгое время оставалась вне поля зрения W3C. (О какой же полнокровной поддержке стиля может тогда идти речь?) Компания Adobe Systems в рамках своей технологии переносимых PDF-документов давно решила эту проблему, причем даже в более строгой постановке - с поддержкой принципа самодостаточности документа. Технология Amber, вошедшая впоследствии в Adobe Acrobat 3.0, не только позволила добиться отображения страницы с соблюдением жестких требований полиграфии, но и за счет оптимизированного бинарного представления обеспечила "интеллектуальную" динамическую постраничную подкачку через Internet. Понимая неизбежность мультиплатформной Web-поддержки шрифтов, в конце апреля Adode и Microsoft анонсировали спецификацию OpenType, которая заложила основы использования Web-шрифтов с учетом особенностей существующих форматов Adobe Type 1 и TrueType. (Реальную работу с OpenType планируется начать в первом квартале 1998 г.) Да и Netscape предложила вполне конкретный механизм, который так и называется "динамические шрифты" (TrueDoc). W3C-спецификация механизма работы с Web-шрифтами (web fonts), которая расширяет трактовки CSS1 и допускает разные форматы шрифтов (Adobe Type 1, TrueType, OpenType и др.), появилась лишь 21 июля этого года, спустя две недели после выхода HTML 4.0 (Web Fonts // W3C Working Draft, 21 Jul 1997).

Читая проект HTML 4.0, в котором информационная часть гораздо больше нормативной, невольно ловишь себя на мысли, что в конечном итоге рабочая группа недалеко ушла от простой констатации того факта, что отныне любой документ HTML - это законный документ прародителя HTML, языка SGML. И горькую пилюлю не в силах подсластить даже известие о том, что в вопросе интернационализации HTML сделан наконец-то прорыв - обеспечивается полнокровная поддержка Unicode 2.0 и соответствующего универсального набора символов (Universal Character Set по стандарту ISO-10646). При этом обработка текста допускает перекодировку в подмножество набора символов данного документа, например, реализацию ее на основе ISO 8859-1 (Latin1, западноевропейские языки), ISO 8859-5 (кириллица)* и др.

Думаю, всем нам далеко не безразлична судьба HTML - на сегодняшний день основного языка общения в Internet, тем более, что на нем создаются (надо думать, не на год и не на два) информационные богатства всемирной Сети. Так что в связи с капитальным ремонтом HTML уместно спросить, а на решение каких, собственно, задач должен ориентироваться этот язык. На представление текста? Тогда почему столь бедны средства верстки? На отображение графики? Отчего все дело ограничивается воспроизведением двух растровых форматов представления (GIF89a и JPEG JFIF) да примитивной GIF-анимацией, когда векторные и композитные форматы (SVF, CGM, PDF) намного лучше подходят для Internet? На сетевую поддержку мультимедиа? Ее нет и в помине (подключаемые модули типа Shockwave к стандартным базовым средствам отнести никак нельзя). На обеспечение гипертекстовых связей (для чего вроде бы и создавался HTML)? Как объяснить тогда отсутствие обратных ссылок, наследования связей, системы отложенной коммутации, атрибутов и типов соединения, структуры узлов адресации и многого другого? Нет даже требования наличия самого элементарного штатного механизма, позволяющего автоматически прописать внутри HTML-документа его координаты - единый идентификатор ресурса (URI, Uniform Resource Identifier), а ведь закладки - вещь внешняя и ненадежная: как потом искать первоисточник?

Только сейчас разработчики HTML начинают задумываться о том, что хорошее отображение на экране - еще не все: Web-документ, оказывается, должен еще и нормально распечатываться на принтере. А как быть со звуковой интерпретацией (при воспроизведении синтезатором речи) и с принудительной визуальной навигацией (по принципу гида-экскурсовода)? Я уже не говорю о персонифицированной обработке Web-страниц. В нынешнем виде HTML больше всего напоминает щенка из мультфильма, который, копируя знакомых зверушек, нацепил на хвост веер, на лапы - ласты, а на голову - щетку и присвоил себе меткое название "павлин-утка-еж". Самое странное и удивительное состоит в том, что W3C до сих пор даже не проводит сертификацию браузеров на соответствие стандарту HTML. Чтобы балансировать на краю совместимости несовместимого, разработчики Web-страниц вынуждены отлаживать свои творения на нескольких версиях разных браузеров. Поддерживаемые для этих целей консорциумом W3C экспериментальные браузеры вроде Amaya (сменившего устаревшую Arena) сильно отстают от декларированных положений W3C, а потому не могут считаться полноценными верификаторами.

Нет общедоступной строго продуманной системы тестов, позволяющей установить отклонения реализации от зафиксированных на бумаге положений (аналогичной той, которая существует, например, для компиляторов языка Ада). Взамен W3C все в том же проекте HTML 4.0 предлагает для проверки корректности Web-документа пользоваться синтаксическими анализаторами языка SGML. Но это лишь синтаксис, да к тому же иного, куда менее строгого языка!

Немного истории

SGML

Давайте, пользуясь принципом колеса обозрения, поднимемся повыше и посмотрим назад - вдаль, в то время, которое предшествовало появлению HTML. Отсчет здесь принято вести с языка SGML (Standard Generalized Markup Language - стандартный обобщенный язык разметки). Что и говорить, в плане описания документа юному отпрыску (HTML) далеко до своего деда, увенчанного в 1986 г. почетной наградой, - Международная организация по стандартизации (ISO), приняв стандарт ISO-8879, включила SGML в число избранных. Зато потомок оказался гораздо проще (сейчас, правда, этого уже не скажешь). SGML - это не просто язык разметки текста, он определяет расширяемое семейство языков. HTML же представляет собой DTD (Document Type Definition), т. е. язык для описания одного вполне конкретного типа документов из допускаемых SGML. Свое распространение HTML получил на заре WWW-революции во многом благодаря простоте освоения и простоте реализации соответствующих интерпретаторов. Тогда более всего требовались незамысловатые средства для сетевых электронных публикаций с возможностью логической навигации по множеству разрозненных документов - своеобразное воплощение мечты Теда Нельсона об открытом пространстве документов (docuverse), в котором все они связаны гипертекстовыми переходами.

Об основополагающих принципах и исторических связях SGML и HTML читатель сможет узнать из статьи Д. Кирсанова "Краткая история HTML" ("Мир ПК", # 4/97, с. 112). Я же затрону здесь историю самого SGML. Идея отделения структуры (structure) информации от содержания (content) на основе формальных грамматик (декларативного языка) давно витала в воздухе. Тридцать лет назад, в сентябре 1967 г., Уильям Танниклифф, председатель Комитета по композиционным решениям (Composition Committee) Ассоциации по графическим коммуникациям (GCA - Graphic Communications Association), предложил провести четкий формальный водораздел между содержанием и форматом представления информации. Примерно в это же время, в конце 1960-х гг., некий дизайнер книг из Нью-Йорка по имени Стенли Райс предложил идею создания универсального каталога на основе параметризованных команд для редакционной структуры. Поняв настоятельную необходимость глубокой проработки идей обобщенного кодирования (generic coding), Норман Шарпф, тогдашний руководитель GCA, принял решение приступить в рамках комитета Танниклиффа к реализации соответствующего проекта. Сформулированная в результате этого концепция GenCode заложила основы будущего стандарта SGML. В 1969 г. американский ученый Чарльз Гольдфарб возглавил работу исследовательской группы в IBM, целью которой была проработка принципов интегрированных информационных систем в области законодательства. Плодом усилий этого коллектива, куда входили также Эдвард Мошер и Реймонд Лори, стал GML - обобщенный язык разметки (Generalized Markup Language). Поговаривают, что его название составлено из начальных букв фамилий трех главных разработчиков (Goldfarb, Mosher, Lorie).

Многие решения этой группы нашли применение в различных издательских системах IBM. Но Гольдфарб не остановился на достигнутом, а продолжил свои изыскания в плане расширения возможностей GML, которые впоследствии вошли в язык SGML. В отличие от своего предшественника, SGML, строго говоря, не был практическим языком, а сразу создавался как промышленный стандарт. Процесс этот длился довольно долго - почти девять лет, начиная от работ Комитета по обработке информации в рамках Американского института стандартов (ANSI) и заканчивая официальным принятием стандарта ISO в 1986 г. Язык SGML в рекордные сроки был взят на вооружение Европейской лабораторией физики частиц (CERN), откуда и начали свой путь Всемирная паутина (WWW) и лежащий в ее основе язык HTML.

С самого своего рождения язык HTML был представлен в текстовом виде, где содержательная информация перемежалась командами разметки (тегами). То есть предполагалось, что творить на нем должен человек. Ныне же очень многие страницы в Internet автоматически генерируются специальными средствами графической компоновки страниц или приложениями баз данных. Технических деталей стало так много, что даже без встроенных стилевых таблиц (которые встречаются пока редко) любая мало-мальски удобная Web-страница на 2/3, а то и на 3/4 состоит из управляющей информации. Исходная идея SGML - отделить зерна от плевел (структуру от содержания) - в ходе эволюции HTML оказалась почти напрочь забытой. Авторы SGML изначально отвергли такие "мелочи", как внешний облик и обратная связь с читателем (им это было попросту не нужно). Как следствие, HTML стал языком, в котором исключения превалируют над правилами.

Дональд Кнут и литературное программирование

Поднимемся на нашем колесе обозрения еще немного выше и взглянем на, казалось бы, не относящиеся к делу работы Кнута. Дональд Кнут, профессор Стэнфордского университета, прославившийся фундаментальным трудом "Искусство программирования", одним из первых обратил внимание общественности на то, что между документом и программой куда больше общего, чем различного. (Если вам еще не довелось ознакомиться с его блестящей лекцией 1974 г. "Программирование как искусство", то обязательно восполните пробел при первом же удобном случае. Она вошла в сборник "Лекции лауреатов премии Тьюринга", выпущенный издательством "Мир" в 1993 г.). Эстетика в программировании (как, кстати, и в любой другой области человеческого знания) в значительной степени способствует появлению качественных и надежных решений. Огромная заслуга Кнута в том, что он не просто декларировал важность эстетического восприятия, а реализовал свою идею, на практике продемонстрировав ее огромные возможности.

Будучи известным специалистом в области комбинаторики и теории графов, Кнут пришел к выводу, что одним из самых удобных способов представления любой программы является не простой текстовый файл, а сеть, паутина (web) из множества взаимосвязанных узлов, формирующих зависимости между элементами программы. Соответствующая инструментальная система с именем WEB сначала была реализована им на Паскале, а затем в 1987 г. Сильвио Леви из Беркли переписал ее и систему TeX на Си (CWEB). Конкретным приложением системы WEB явилась среда поддержки разработанного в 1983 г. все тем же Кнутом TeX - языка представления сложных документов, который и по сей день активно используется в научном мире для подготовки насыщенных логическими связями трудов и монографий с замысловатыми математическими формулами.

Кнут стал основоположником новой компьютерной дисциплины, которой дал имя literate programming (см. Donald E. Knuth. Literate Programming // The Computer Journal, 1984, Vol.27, # 2, pp. 97-111). Буквально это "грамотное", "образованное" или "интеллигентное программирование", но мне больше по душе перевод "литературное программирование". В понимании Кнута написание хорошей программы как с творческой, так и с эстетической точки зрения мало чем должно отличаться от труда поэта, литератора. Дональд Кнут тонко уловил, что представление исходного кода программы в виде особого гипертекста с удобным шрифтовым оформлением не просто элегантно - оно существенно уменьшает число ошибок и повышает производительность труда программиста. Недаром вслед за WEB и CWEB появились реализации для поддержки помимо Паскаля и Си еще и других языков - АПЛ, Фортрана, Лиспа, Scheme, Java. Идя по стопам Кнута, Норман Рамсей даже создал не зависящую от языка систему структурированной документации с именем SpiderWEB. Так что нельзя не признать, что Дональд Кнут создал свою "паутину", причем сделал это раньше, лучше и глубже, чем "отцы" WWW. В предисловии к своей книге (Donald E. Knuth. Literate Programming // Stanford, CSLI Lecture Notes, N 27, 1992) Кнут с плохо скрываемой досадой отметил: "Я использовал слово WEB для этой цели гораздо раньше, чем его позаимствовал швейцарский CERN". (Проект World Wide Web был начат Тимом Бернерсом-Ли в 1989 г.)

Проецируя идеи Кнута на Internet, можно сказать, что накопление и переосмысление информационного богатства человечества в электронном виде - всего того, что ныне называют модным словечком "контент" (content, т.е. "содержание"), - должно строиться прежде всего на принципах литературного (или, если хотите, художественного, эстетического) программирования. Что же касается нынешних, подчас сумбурных попыток разрешения возникших проблем, то наиболее примечательными, хотя и разными по своему масштабу и степени реализации, являются две тесно связанные между собой инициативы:

- исследовательская, Meta Content Framework ("каркас метаконтента"), разработанная группой ATG в исследовательском центре Apple (http://mcf.research.apple.com/mcf.html); сейчас эти идеи активно развивает Netscape, получив у Apple соответствующую лицензию;
- нормативно-методическая, OII (Open Information Interchange - открытый обмен информацией; см. http://www2.echo.lu/oii/en/docstand.html), являющаяся ныне составной частью глобальной европейской программы INFO 2000; среди ее стандартов наиболее интересны DSSSL и XML.

Хронология развития Web-языков

I - основные языки обобщенной разметки
II - вспомогательные языки
III - сопутствующие языки и спецификации

I	II	III	год	организация
GenCode			1967	GCA
GML			1969	IBM
SGML			1986	ISO
HTML			1991	CERN
HTML 1.2			1993	CERN
HTML 2.0			1995	W3C
		Java	1995	Sun
		JavaScript	1995	Netscape
		CML	1995	Nottingham University
		Meta Content Framework	1996	Apple
	HyTime		1996	ISO
	DSSSL		1996	ISO
		HCML	1996	The Word Electric и др.
	CSS		1996	W3C
HTML 3.2			1997	W3C
		Dynamic HTML	1997	Microsoft
		dynamic HTML	1997	Netscape
		JASS	1997	Netscape
HTML 4.0			1997	W3C
XML			1997	W3C
		XML-Data	1997	Microsoft
		MCF	1997	Netscape
	MCL		1997	W3C
		MathML	1997	W3C
		CDF	1997	Microsoft
		OFX	1997	Microsoft, Intuit, CheckFree
	XSL		1997	Microsoft
		OSD	1997	Microsoft, Marimba
	RDF		1997	W3C

* Кодировка ISO 8859-5 используется на русскоязычных страницах Internet значительно реже, чем КОИ-8 и кодировка Windows (CP 1251), но это международный стандарт, который поддерживают практически все ведущие производители программного и аппаратного обеспечения: IBM, DEC, Hewlett-Packard, Sun Microsystems, Oracle, Informix, Netscape Communications (особую позицию заняла только Microsoft). ISO 8859-5 (в отличие от КОИ-8, существующей в вариантах для русского и украинского) является универсальной кодировкой кириллицы, т. е. включает буквы русского, украинского, белорусского, болгарского, сербского и македонского алфавитов.