

Інфокомунікаційні системи

УДК 004/7

А.С. Гордиенко

Харьковская государственная академия культуры, Харьков

СРАВНИТЕЛЬНЫЙ АНАЛИЗ КЛИЕНТСКИХ ВЕБ-ТЕХНОЛОГИЙ И ТЕХНОЛОГИЙ НАСТОЛЬНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ: ИСТОРИЯ И СОВРЕМЕННЫЕ ПРОБЛЕМЫ

В статье рассматривается история технологического развития клиентской части веб-приложений по мере приближения их к настольным приложениям, приводится краткий обзор проблем соответствующих технологий и причин их возникновения, дается оценка современному состоянию индустрии.

Ключевые слова: настольное программное обеспечение, веб-технологии, HTML, HTML5.

Введение

За последние 20 лет Интернет и настольное программное обеспечение прошли путь от нишевого решения, используемого отдельными профессионалами и энтузиастами, до неотъемлемой части жизни людей в большинстве стран. Соответствующие технологии стремительно развивались во всех аспектах, зачастую без общей организации, на эмпирической базе. В статье предпринята попытка вкратце описать историю развития этих технологий, рассмотреть их преимущества и недостатки, представить свой взгляд на текущее положение вещей и описать видение ближайшего будущего этой сферы.

Развитие веб-технологий и технологий настольного программного обеспечения

В конце 1990-х годов типичная веб-страница представляла собой текст с гиперссылками и навигационное меню с небольшими изображениями – это была, фактически, гипертекстовая версия газетной страницы. Язык HTML на тот момент полностью удовлетворял техническим потребностям Интернет-паутины. Настольные и веб-приложения радикально отличались друг от друга как по функциям, так и по внешнему виду.

Однако в процессе развития сети появилась потребность в динамическом, интерактивном содержании веб-страниц и более сложном форматировании и оформлении, чем мог предоставить HTML. Получило распространение динамическое формирование страниц на стороне сервера с помощью таких технологий, как CGI, Perl, ColdFusion, PHP, ASP. Для обработки простых интерактивных сценариев на стороне пользователя в браузеры была добавлена поддержка JavaScript. Сложная верстка и оформле-

ние средствами HTML оказались практически невозможными, и для описания внешнего вида стал применяться язык CSS. Отсутствие клиентского хранилища данных компенсировали с помощью HTTP Cookie. Ни одна из этих технологий не предусматривалась стандартом HTML, они применялись в качестве решения практических ограничений на том или ином этапе.

Разработчики, фактически, пытались приблизить все более сложные веб-страницы (которые теперь уже принято называть веб-приложениями) к уровню настольного программного обеспечения по времени отклика и скорости работы, функциональности и возможностям интерактивного взаимодействия, богатству оформления. Но технологические ограничения (в первую очередь – необходимость работать через удаленные интерпретаторы) затрудняли эту задачу.

Следует отметить, что при развитии веб-части Интернет постоянно был и остается актуальным вопрос совместимости страниц с различными браузерами и платформами. В то время как базовые стандарты (HTML, JavaScript, CSS) едины, тонкости их реализации и конкретные алгоритмы рендеринга страниц в разных системах могут отличаться.

В качестве решения проблемы совместимости веб-приложений, интерактивности и задействования ресурсов клиентского компьютера компания Sun предложила апплеты Java – программы, которые выполняются в окне браузера как часть веб-страницы с помощью виртуальной машины Java, доступной на большинстве платформ. Java-апплеты не получили повсеместного распространения, в первую очередь, из-за сложности создания привлекательного пользовательского интерфейса. Появились также ориентированные на медиа-контент встраиваемые в веб-страницы технологии, такие как

Shockwave, QuickTime и RealMedia. Наибольшего же успеха удалось достичь платформе Macromedia Flash, которая сегодня установлена на более чем 99% настольных компьютеров [1], и используется на четверти веб-сайтов Интернет [2]. Flash предоставляет одновременно отличный инструментарий для создания анимации и интерфейсов (в частности, с помощью смежной технологии Flex), встроенный скриптовый язык с развитыми библиотеками, возможности по работе с аудио и видео и универсальный плагин-плеер для выполнения Flash-программ. Компания Microsoft имеет собственную технологию Silverlight, во много аналогичную Flash.

Для обмена данных между веб-страницей и веб-сервером традиционно необходимо было перезагружать страницу полностью, что вызывало существенную временную задержку и повторный рендеринг. В 2005 году после обратной инженерии веб-приложений компании Google стал широко известен метод, который позволял в асинхронном режиме обмениваться сообщениями с браузером без перезагрузки страницы – так называемый Ajax. Такой подход позволил существенно приблизить интерактивность интерфейса веб-приложений к настольным программам. В то же время, подход был основан на использовании нестандартного расширения JavaScript, что впоследствии привело к усложнению этого изначально простого языка, появлению дополнительных стандартов (JSON, W3C XMLHttpRequest) и специализированных библиотек (jQuery, Ext JS и т.п.). В результате сегодня JavaScript-программирование является отдельной специальностью, достаточно сложной в освоении.

В последние несколько лет получили распространение мобильные операционные системы нового поколения – в первую очередь, iOS и Android. Основные приложения этих систем обладают свойствами настольных приложений (выполняются локально и активно используют аппаратные ресурсы), однако активно используются и веб-приложения, запускаемые как через встроенный браузер, так и наравне с обыкновенными приложениями. По оценкам Morgan Stanley, к 2015 году веб будет чаще просматриваться на мобильных устройствах, чем на персональных компьютерах и ноутбуках [3].

Слабая техническая реализация Flash-плеера и постоянное выявление уязвимостей в различных его реализациях ограничило возможности развития этой технологии.

Одним из факторов, которые не позволили Flash стать доминирующей интерактивной платформой Интернет-паутины, стал отказ компании Apple поддерживать ее в браузерах устройств iPhone и iPad. В свете последних новостей [4], можно констатировать спад интереса ко всем встраиваемым в веб-страницы технологиям.

Будущее развития веб-приложений большинство компаний-лидеров индустрии видит в HTML5. В новой версии языка предусмотрена поддержка ряда новых спецификаций, расширяющих возможности языка в области отображения 2D и 3D графики и анимации, медиа-контента, геолокации, микроформатов, офлайн-режима страниц, drag-and-drop, поддержки файловой системы и других. Очевидно, веб-приложения при условии внедрения HTML5 сделают еще один шаг к уровню настольного программного обеспечения [5].

Тут следует отметить, что по сути, веб-приложения и настольные приложения выполняют одну и ту же задачу: предоставить информацию и функциональность на экран пользователя. Главное отличие состоит в том, что большая часть логики и данных веб-приложения расположены на сервере. Это позволяет, с одной стороны, избежать процессов установки и обновления, а с другой – делает пользователя зависимым от доступности сервера, требует его идентификации и работы через программу-посредника, браузер. На примере Apple AppStore и MacStore можно видеть, что для настольного программного обеспечения также возможно усовершенствовать и упростить схему доставки функциональности пользователю.

Как видно, на протяжении долгих лет HTML развивался экстенсивным образом. На данный момент язык оброс большим количеством побочных технологий с различной логикой и инструментарием. Возникает вопрос – каким образом технология начала 90-х на протяжении 20 лет занимает центральное положение в Интернет-паутине, сплошь наполненной инновациями?

Во-первых, следование единой технологии обусловлено требованием обратной совместимости. Веб-рынок достаточно сильно диверсифицирован и единственный действующий контролирующий орган W3C не имеет достаточного влияния и ресурсов для революционного внедрения новой, отвечающей современным реалиям Интернет технологии. Для сравнения, принадлежащие отдельным компаниям технологии Flash и Silverlight активно развивались, теряя от версии к версии полную обратную совместимость на уровне кода (что компенсируется сохранением обратной совместимости на уровне виртуальной машины). Также сложность технологий вызывает инертность пользователей, специалистов и организаций, которым непросто будет перестроиться на новые модели.

Во-вторых, сказывается дефицит качественных универсальных альтернатив HTML. Интернет-паутина – это, бесспорно, наиболее развитая реализация идеи гипертекста, и самая массовая система предоставления функциональности. Из технологий помимо «стека» HTML, которые переносимы между

основними платформами для клієнтських систем (Windows, Mac/OSX і Unix) і мобільними пристроями (iOS, Android), можна назвати тільки мови C/C++. Определенные возможности предоставляют Adobe AIR (прекращена підтримка Unix), Java (не підтримується на iOS), Microsoft .NET (роботають вне Windows тільки через зв'язку Mono/MonoTouch), но по большей части, единственное универсальное решение на сегодня – веб-приложение.

И, наконец, важнейшая причина заключается в коммерческом успехе Интернет-паутины. Стремительно развившаяся Интернет-индустрия привлекла огромное количество средств, специалистов и пользователей.

Концентрация человеческих потоков и данных на более успешных веб-сайтах привела к образованию крупнейших корпораций, таких как Google, Amazon, Yahoo!, eBay, Facebook. Современная структура веб-технологий, несмотря на очевидные минусы, устраивает большинство крупных игроков, которые могли бы содействовать инновациям. В частности, отсутствие встроенной в Интернет-паутину модели поиска, обеспечивает Google более миллиарда просмотров страниц ежедневно, а реклама веб-сайтов приносит 96% прибыли [6]. Другие компании используют концентрацию информации в своих руках, обусловленную клиент-серверной архитектурой веб-сайтов, в коммерческих целях; так, например, социальная сеть Facebook использует личные данные пользователей для таргетирования рекламы сети Microsoft.

Централизацию можно назвать главенствующим трендом всей IT-индустрии последней пары лет; несмотря на экспоненциальный рост производительной мощности и дискового пространства клиентских станций, «облачные» технологии внедряются повсеместно, принося, с одной стороны, удобство использования из любой точки планеты, а с другой – проблемы с безопасностью данных и нагрузку на сетевую инфраструктуру [7].

Выводы

Таким образом, в ближайшем будущем можно ожидать дальнейшего укрепления позиций веб-ориентированного программного обеспечения. Сложность и требования к производительности клиентской части веб-приложений будут далее возрастать. Долгосрочный успех развития веб-технологий будет обусловлен тем, насколько удобными окажутся средства создания приложений на основе HTML5.

Список литературы

1. Adobe Flash Platform Statistics, PC penetration. [Электронный ресурс] // Adobe.com. – Режим доступа к ресурсу: <http://www.adobe.com/products/flashplatformruntimes/statistics.html>. – Загл. с экрана.
2. Usage of client-side programming languages broken down by web servers. [Электронный ресурс] // W3Techs Web Technology surveys. – Режим доступа к ресурсу: http://w3techs.com/technologies/cross/client_side_language/web_server. – Загл. с экрана.
3. Morgan Stanley. Internet Trends 2010 [Электронный ресурс] // Morgan Stanley. – Режим доступа к ресурсу: http://www.morganstanley.com/institutional/techresearch/pdfs/Internet_Trends_041210.pdf. – Загл. с экрана.
4. Flash, Silverlight and the end of the line for browser plug-ins [Электронный ресурс] // GeekWire. – Режим доступа к ресурсу: <http://www.geekwire.com/2011/adobe-flash-microsoft-silverlight-sunset-browser-plugin>. – Загл. с экрана.
5. Kadlec T. Implementing Responsive Design: Building sites for an anywhere, everywhere web / Tim Kadlec // Voices That Matter. – Berkeley, CA: New Riders Press, 2012. – P. 227-254.
6. Google Quarterly Report on Form 10-Q, Q3 2011 [Электронный ресурс] // US Securities and Exchange Commission. – Режим доступа к ресурсу: <http://www.sec.gov/Archives/edgar/data/1288776/000119312511282235/d228523d10q.htm>. – Загл. с экрана.
7. Sosinsky B. Cloud Computing Bible / Barrie Sosinsky. – Indianapolis, NA: Wiley, 2011. – 528 p.

Поступила в редколлегию 1.08.2012

Рецензент: д-р техн. наук, проф. И.В. Гребенник, Харьковский национальный университет радиоэлектроники, Харьков.

ПОРІВНЯЛЬНИЙ АНАЛІЗ КЛІЄНТСЬКИХ ВЕБ-ТЕХНОЛОГІЙ І ТЕХНОЛОГІЙ НАСТІЛЬНОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ: ІСТОРІЯ ТА СУЧАСНІ ПРОБЛЕМИ

А.С. Гордієнко

В статті розглянуто історію технологічного розвитку клієнтської частини веб-додатків в міру наближення їх до настільних додатків, наводиться короткий огляд проблем відповідних технологій та причин їх виникнення, дається оцінка сучасного стану індустрії.

Ключові слова: настільне програмне забезпечення, веб-серверу-технології, HTML, HTML5.

COMPARATIVE ANALYSIS OF CLIENT WEB TECHNOLOGIES AND TECHNOLOGIES OF DESKTOP SOFTWARE: HISTORY AND CURRENT ISSUES

A.S. Gordiyenko

The article provides a brief coverage of web technologies history and their relation to desktop applications, gives an overview of issues with related technologies and their causes, assesses the current state of the industry.

Keywords: table software, to the web-technology, HTML, HTML5.