

К. И. Прокопенко, А. И. Прокопенко

## СОВРЕМЕННЫЕ ВЕБ-СИСТЕМЫ И ПРОБЛЕМЫ БЫСТРОДЕЙСТВИЯ КЛИЕНТСКИХ ПРИЛОЖЕНИЙ

В веб-приложениях нового поколения особое внимание уделяется степени удовлетворенности пользователя работой с приложением. В первую очередь идет речь об интерактивности приложения, во вторую очередь – о его производительности. Производительность оттесняется на второе место, уступив место удобству работы пользователя, поскольку в настоящее время поиск высокопроизводительной вычислительной машины для обеспечения работы той или иной системы, использующейся в малом и среднем бизнесе, не является проблемой. Но при попытке обеспечить должный уровень быстродействия взаимодействия с пользователем, разработчики сталкиваются с определенным числом вопросов. Поскольку речь идет о моделировании системы, которая по своей природе является веб-приложением, то возникает ряд вопросов, связанных с тем, каким образом обеспечить пользователю должный уровень «реактивности» приложения? Актуальность данного вопроса особенно повышается, если рассмотреть моделируемый веб-проект как продукт, который будет использоваться большим числом людей, которые в конечном итоге определяют степень удобства его использования, а в итоге и экономическую эффективность.

Проблема формулируется следующим образом: как обеспечить качество взаимодействия веб-проекта с конечным пользователем на уровне windows forms приложения? В настоящее время существуют несколько возможных подходов к решению данной задачи. Суть всех методов сводится к следующему: необходимо устранить основную причину замедления взаимодействия пользователя с веб-системой, - избавиться от постоянных перезагрузок страницы при обратной отправке данных на сервер. Данный шаг во-первых экономит время и нервы пользователя, во-вторых экономит тот объем информации, который будет передан по сети. Передача данных практически во всех методах происходит асинхронно, с помощью объекта XMLHttpRequest, который анонсировала фирма Microsoft в 2000г. Из существующих приемов повышения реактивности клиентских

приложений, следует выделить три метода: 1. Использование технологии AJAX, 2. Использование технологии Flash, 3. Использование технологии Silverlight. Все три подхода основываются на одном и том же принципе. Единственным отличием является то, какой набор инструментов программирования предоставляется для обеспечения необходимой функциональности на клиентской стороне. AJAX – это просто пересылка данных, не более и не менее. Поэтому в случае с AJAX технологией, вся обработка данных ложится на JavaScript, что само по себе является весьма неудобным, даже если учесть, что данные возвращаются в JSON нотификации и представляют из себя готовые JavaScript-объекты, поскольку возможности JavaScript в контексте веб-приложений довольно скромные и ограничиваются работой с DOM и работой с ActiveX-объектами. В отличие от обычного AJAX, Flash и Silverlight предоставляют богатый набор инструментов, которыми может пользоваться разработчик на стороне клиентской части веб-приложения. Данная возможность обеспечивается «толстым клиентом», через которого работают обе технологии. Другими словами, возможности JavaScript расширяются с помощью дополнительных библиотек, add-on-ов и ActiveX-объектов.

Ключевым компонентом стратегии разработки веб-приложений следующего поколения корпорации Майкрософт является новая технология Microsoft® Silverlight™, которая прежде имела кодовое название «WPF/E». Предназначенная для использования возможностей Windows® Presentation Foundation в Интернете технология Silverlight предоставляет богатое содержимое, обеспечивающее эффект присутствия и поддерживающее все остальные среды разработки веб-приложений, включая AJAX ASP.NET.

Чтобы охватить веб-сообщество целиком, Silverlight должна успешно выполняться на ряде популярных операционных систем и в популярных обозревателях. В результате первый выпуск поддерживает обозреватели Firefox и Safari для операционной системы Mac OS X, а также обозреватели Firefox и Internet Explorer® для операционной системы Windows. Поддержка других операционных систем и обозревателей будет добавлена в дальнейшем. В добавление к этим возможностям Silverlight также является

полностью самодостаточной и не зависит от других продуктов, например от проигрывателя Windows Media® для воспроизведения видео и платформы Microsoft .NET Framework 3.0 для преобразования XAML.

По сути Silverlight является подключаемым модулем обозревателя для обработки XAML, представляющим обозревателю внутреннюю модель DOM и модель событий с помощью сценариев. Поэтому проектировщик может создать документ XAML с графикой, анимацией и диаграммами, а разработчик может подключить этот документ к коду на странице для обеспечения функциональных возможностей. Поскольку язык XAML основан на XML, документ, определяющий загружаемый клиенту пользовательский интерфейс, является текстовым и поэтому не вызывает конфликтов с поисковыми системами и брандмауэрами. Кроме того, XAML может собираться и создаваться серверным приложением во время выполнения, что предоставляет не только насыщенный графический, но и динамический интерфейс с широкими возможностями настройки.

Структура простого приложения Silverlight, использующего статический файл XAML для определения пользовательского интерфейса и JavaScript для обработки событий, показана на рис. 1.

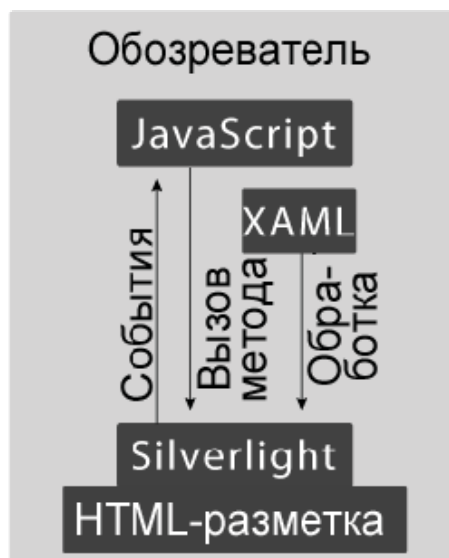


Рисунок 1 – Структура простого приложения Silverlight

Обозреватель создает экземпляр подключаемого модуля и в качестве части этого процесса загружает файл XAML. События в этом файле, например, нажатие кнопки, перехватываются обозревателем и

обрабатываются JavaScript. Благодаря представлению модели DOM содержимого Silverlight JavaScript может также динамически обновлять содержимое Silverlight, изменяя состояние сформированного содержимого.

Архитектура, поддерживающая приложение Silverlight, показана на рис. 2. Основным интерфейсом программирования является интерфейс API модели DOM JavaScript. Это позволяет отвечать на события, вызванные в XAML Silverlight (например, по завершении загрузки содержимого или окончании воспроизведения анимации). Кроме того, можно вызывать методы для управления презентацией (например, запуска анимации или приостановки воспроизведения видео). Подо всеми этим лежит механизм синтаксического анализа XAML. Анализатор создает DOM XAML в памяти для использования ядром презентации, которое управляет формированием графики и анимации, определенных XAML. Кроме того, модуль времени выполнения содержит кодеки, необходимые для воспроизведения мультимедийного содержимого в форматах WMV, WMA и MP3.

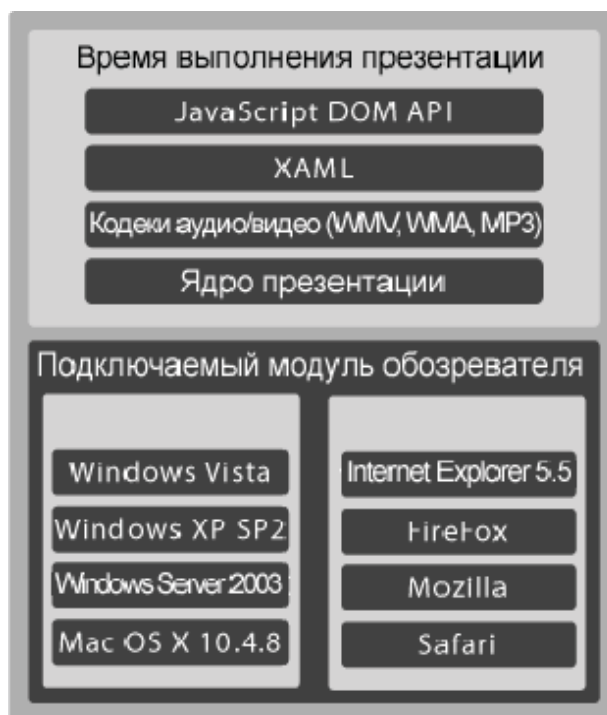


Рисунок 2 – Архитектура Silverlight

Наконец, модуль времени выполнения содержит ядро презентации, управляющее процессом визуализации. Этот модуль времени выполнения для презентации встроен в подключаемый

модуль обозревателя, поддерживающий некоторые версии Windows и Mac OS X с помощью указанных ранее обозревателей. Конечным результатом является самодостаточный модуль отображения графики и мультимедиа, который может подключаться к обозревателю и программироваться с помощью сценариев JavaScript.

Едва ли можно думать о Silverlight как только о клиентской технологии предназначенной для встраивания насыщенного содержимого в обозреватель. Такое насыщенное содержимое уже возможно благодаря использованию закрытых подключаемых модулей, построенных в виде апплетов Java, элементов управления ActiveX® или приложений Flash. Silverlight, напротив, является открытой технологией в смысле того, что пользовательский интерфейс определен в текстовом коде XAML, а программируемость достигается путем использования JavaScript. Это позволяет разработчикам легко создавать приложения, взаимодействующие с серверами, на которых расположены данные.

С выходом новой версии технологии Silverlight 2.0, разработчики получили массу новых возможностей для реализации своих амбициозных идей.

Silverlight 2.0 – это в первую очередь сверхкомпактный .NET Framework, обладающий самыми высокими ограничениями по безопасности за всю историю .NET. Код можно писать на C# 3.0 и использовать LINQ! Иногда Silverlight определяют так: "частное производное WPF" ("WPF subset"). Это некорректно, хоть Silverlight и содержит в себе компактную версию WPF, это лишь библиотека для вывода информации, т.е. в первую очередь нужно рассматривать Silverlight как платформу, хотя "WPF subset" и является главной составляющей Silverlight, без которой просто не было бы смысла его использовать.

Silverlight включает в себя мини-версию WPF. В ней нет поддержки трёхмерной графики, однако трансформации, набор элементов управления и графических элементов весьма внушительные.

Так же как и у "большого брата", презентационный движок Silverlight основан на XAML. XAML способен сэкономить компаниям, специализирующимся на разработке ПО внушительное количество времени и, соответственно, денег, позволив так же, опираясь на свою

платформу – WPF или Silverlight, делать потрясающие эффекты. Во-первых, для XAML есть достаточно удобная программа для дизайнеров – Microsoft Expression Blend. Т.е. вместо обычного сейчас процесса, когда дизайнер рисует макет, например, в Adobe Illustrator, а потом верстальщик (или программист) верстает HTML и программист добавляет логику, с использованием технологии Silverlight программисту уже будет приходиться получать готовый XAML, который осталось лишь чуточку поправить в соответствии с логикой. Во-вторых, возможности у XAML гораздо-гораздо шире, чем у того, что используется сейчас большинством – Windows Forms и HTML. Т.е. вещи, которые в HTML или Windows Forms ранее было делать либо крайне сложно либо даже невозможно, теперь можно будет делать с лёгкостью. В-третьих, XAML единый и для Desktop-приложений, и для Веб-приложений, что в свою очередь создает лёгкость в использовании одного макета страницы как в WPF приложении, так и в Silverlight приложении.

Главные конкуренты Silverlight 2.0 – это Adobe Flash и Sun JavaFX. Среди .NET разработчиков уже сейчас есть некоторая прослойка, использующая связку Flex/Flash/ASP.NET в своих проектах. Однако мне кажется, что Silverlight 2.0 объективно стоит выше по технологическому уровню, и Flash, изначально ориентированный на мультики-рекламные ролики, не сможет удержать RIA разработчиков в своих объятиях. Путь технологии Silverlight в Интернете только начинается. С богатыми возможностями механизма визуализации XAML и гибкостью использования стандартных технологий веб-разработки Silverlight заслуживает того, чтобы быть добавленной в набор инструментов разработчика.

#### ЛИТЕРАТУРА

1. Laurence Moroney. Introducing Microsoft Silverlight 1.0. – Microsoft Press, 2008 - 225с.
2. Джеффри Рихтер. Программирование на платформе .NET Framework – М. Русская редакция, 2003 -512с.
3. <http://silverlight.net/>
4. <http://www.microsoft.com/rus/msdn/articles/silverlight/>
5. <http://wynapse.com/silverlight/tutor/>