

Pavlo Bazilinskyy

Mikkeli University of Applied Sciences, Finland

Microsoft Silverlight. Overview of the new way of building RIA

Microsoft Silverlight, a new technology from Microsoft is aimed for creation of RIA, which stands for Rich Internet Applications. These applications are web-based solutions containing media elements such as high-definition video, audio etc. One can think of RIA as being similar to rich desktop applications, that are often characterized as being sophisticated solutions that include complex elements of User Interface (UI), such as 3-D graphical elements and graphical elements presented using high definition textures. It was designed to fill the gap between desktop and web programs in terms of creating Graphical User Interfaces (GUI). [1]

Microsoft Silverlight is distributed as a browser plugin of approximately 6MB in size available for most modern browser programs for Microsoft Windows and Mac OS powered computers and, therefore, is a cross-platform cross-browser solution. It is worth mentioning that Linux-based computers are also supported by a partially 3rd party solution Moonlight developed by Microsoft in cooperation with Novell corporation. [2]

This technology was being created having Windows Presentation Foundation (WPF), application development framework developed to enhance creation of projects that use advanced elements of UI such as 3-D graphics, complex animations, hardware acceleration, as a base in its early stages of development. Silverlight and WPF, therefore, have numerous features in common: they both use XAML, eXtended Application Markup Language, which is a new declarative programming model from Microsoft, for design purposes, to describe elements of UI in browser-based projects, .NET platform has been chosen for these 2 technologies as a base for the programming part of development of projects written in either of aforementioned technologies from Microsoft. Additionally to this point, Silverlight at its current stage of progress will not support several features available in WPF such as hardware-

based video acceleration, and full document support, in order to serve Silverlight's cross-platform, cross-browser approach. [3]

RIAs built with Silverlight are run as client-side applications, which does not require a user to refresh the browser to update the application's UI. Noticeably, Silverlight does not have an issue of "Back" button functionality of going back to the previous website and not the previous page, which is considered by many as a big flaw in usability of such solutions in area of creating RIA as Adobe Flash [4]. Silverlight created application often require extensive connections to remote servers and services and, therefore, Silverlight platform provides sophisticated tools for creating solid and reliable links to remote locations. It can run asynchronous communications with the server, allowing the UI to continue to function while waiting for the server response. Moreover, Silverlight applications can be used "Out-of-Browser", which means that they can be downloaded from the server where they are hosted and later be used by a user in offline mode. [1]

At its first iteration Silverlight was created as a purely AJAX and JavaScript based technology where all the code in projects had to be written entirely in JavaScript and XAML. With Silverlight 2 this platform received support of managed code thanks to a limited version .NET runtime being installed on a user's machine. This allowed .NET programmers to write code to be executed on the client PC, and provided a better user experience to users. To add, there were security restrictions built in to it so that the code had limited access to the client computer. With Silverlight 2 programmers can still implement functionality using JavaScript, however, one can also implement the logical part of the project using C#, Visual Basic, Python, Ruby, and managed JavaScript in projects developed using the 2nd version of Silverlight. Dynamic languages, such as IronPython and IronRuby are also supported [6]. Also, XAP packaging system was implemented, which allowed distribution of XAML packages coded as simple ZIP-formatted archives. Silverlight 3 and 4 were essentially additions to their predecessors and instead of providing significant enchantments these 2 releases were focused on adding minor tweaks and fixes to the platform. Silverlight 5 is currently being developed by Microsoft.



Figure: Deep Zoom technology demo

One of the most noticeable features in a current set of tools in Silverlight is Deep Zoom technology. It allows users to pan around and zoom in a large, high resolution image or a large collection of images. It reduces the time required for initial load by downloading only the region being viewed and/or only at the resolution it is displayed at. Subsequent regions are downloaded as the user pans to (or zooms into them); animations are used to hide any jerkiness in the transition [5]. Also, adaptive streaming technology was included, which is, essentially, similar to Deep Zoom in its idea: it allows smooth transactions between different rates of quality of video that is intended to be shown using an embed into a web page video player.

Microsoft developed Silverlight to be as highly platform-independent as possible. Thus, it has significant advantage in performance over Adobe Flash, Java FX and other technologies. By using benchmarking tools such as Bubblemark (available at <http://www.bubblemark.com>), one can notice that Silverlight of version

3 has an advantage in performance over other technologies available for testing by 20-30%.

There are reasons to believe that AJAX migration is probably a matter of time. Silverlight technology can be used to enhance or provide a new approach to exiting projects that were developed using AJAX technology. It is important to understand what is run on a server and what is handled by a client, since these 2 technologies have a core difference in this area. AJAX and Silverlight can cooperate and providing such cooperation is a possible solution for building future RIA. [6]

Literature:

1. **Laurent Bugnion**. Silverlight 4 Unleashed. Sams - October 31, 2010.
2. **Laurence Moroney**. Microsoft Silverlight 4 Step by Step. Microsoft Press - June 30, 2010.
3. **Pete Brown**. Silverlight 4 in Action. Manning Publications - September 27, 2010.
4. **Ashish Ghoda**. Introducing Silverlight 4 (Expert's Voice in Silverlight) . Apress - July 13, 2010.
5. **Liveside**. Silverlight 2 and Deep Zoom – Get ready to build your own photozooming collections. Referred 21.3.2011. Available in www-format: <URL: <http://www.liveside.net/2008/03/06/silverlight-2-and-deep-zoom-get-ready-to-build-your-own-photozooming-collections/>>.
6. **Matthew MacDonald**. Pro Silverlight 4 in C#. Apress - November 9, 2010.