

Opinnäytetyö (AMK)

Tietotekniikan koulutusohjelma

Sulautetut järjestelmät

2010

Jevgeni Suni

WEB 2.0

TECHNOLOGIES AND APPLICATIONS



TURUN AMMATTIKORKEAKOULU
TURKU UNIVERSITY OF APPLIED SCIENCES

OPINNÄYTETYÖ (AMK) | TIIVISTELMÄ

TURUN AMMATTIKORKEAKOULU

Tietotekniikan koulutusohjelma | Sulautetut järjestelmät

Joulukuu 2010 | 40 sivua

Tiina Fern

Jevgeni Suni

WEB 2.0 TECHNOLOGIES AND APPLICATIONS

Eräs tärkeimmistä tämän työn tavoitteista on kuvailla Web 2.0 teknologiat ja sovellukset samalla vertaillen sitä Web 1.0:aan. Selittäen miten jo olemassa olevat Web 1.0:n teknologiat, kuten CSS, XML ja JavaScript voivat yhdistettynä saada aikaan interaktiivisempaa sisältöä nettisivuille. Käyttäen teknologioita kuten Adobe Flex ja Microsoft Silverlight voidaan luoda käyttöjärjestelmästä ja nettiselaimesta riippumattomia verkkosovelluksia edistyksellisellä graafisella käyttöliittymällä.

Tämän työn tarkoituksena on kuvailla Web 2.0 teknologioiden ja sovelluksien etuja, rajoitteita ja vaaroja. Web 2.0:n laajoista sovellusmahdollisuuksista johtuen, kyseisestä opinnäytetyöstä voivat hyötyä IT ammattilaiset, poliitikot, liikemiehet sekä akateemisella alalla työskentelevät.

Opinnäytetyöni on saanut vaikutteita henkilökohtaisesta kokemuksestani Web 2.0 teknologioiden ja sovelluksien käytöstä työelämässäni. Käyttökokemuksieni lisäksi kirjoituksessa on käyty aiheeseen liittyvää suomalaista ja englantilasta kirjallisuutta.

Työssäni käsitellen Web 2.0:n avoimuutta sekä tapaa jolla Web 2.0 hallitsee verkkosisältöä. Esittelen myös Web 2.0 teknologioita, jotka soveltuvat sosiaalisen verkostoitumisen, folksanomiaan, tiedostojen jakamiseen ja hakusanoilla merkkäämiseen (tagging), sosiaalisten hakukoneiden, fuusauksien (mashup) ja pikaviestipalvelujen luomiseen sekä sosiaaliseen suodatukseen (collaborative filtering).

Tämän työn neljännessä luvussa käsitellään Web 2.0:n turvallisuuteen liittyviä kysymyksiä ja ongelmia.

Käsitlemme lisäksi Web 2.0 teknologioiden soveltamista liiketoimintaan ja markkinointiin. Saadaan selville myös miten ”pitkän hännän” teoriaa (”long tale” theory) on hyödynnetty Web 2.0 teknologioiden avustuksella.

ASIASANAT:

Web 2.0, XML, JSON, RSS, Atom, CSS, DOM, JavaScript, AJAX, Adobe Flex, Microsoft Silverlight, security

BACHELOR'S THESIS | ABSTRACT

UNIVERSITY OF APPLIED SCIENCES

Information Technology | Embedded Systems

December 2010 | 40 pages

Tiina Ferm

Jevgeni Suni

WEB 2.0 TECHNOLOGIES AND APPLICATIONS

One of the main purposes of this thesis is to explain technologies and applications of Web 2.0; introducing Web 2.0 and comparing it to Web 1.0; explaining how existing technologies of Web 1.0 such as CSS, XML and JavaScript combined could create more interactive content for websites. Technologies such as Adobe Flex and Microsoft Silverlight could be used to create operation system and browser independent web application with advanced graphic interface.

By explaining advantages, disadvantages and dangers of Web 2.0 technologies and applications, this thesis intends to benefit IT professionals, political, academic and business interest, which profit from the use of these technologies in their professional endeavors.

Research has been influenced by my personal, long lasting experience of using Web 2.0 technologies during my professional and academic undertakings. In addition topic-related secondary literature has also been applied in understanding and explaining applications, technologies and security issues.

This study will also deal with the openness and the way of managing online content in Web 2.0 which has increased users' participation online. Most popular of such Web 2.0 applications are Wikis, blogs and podcasts.

The thesis will introduce sociable technologies of Web 2.0 such as social networking services, social bookmarking, folksonomies, file sharing and tagging, mashups, social search engines, instant messaging and collaborative filtering.

In addition, the chapter number three will overview security related issues and problems of Web 2.0.

Moreover, the applicability of Web 2.0 technologies in business and marketing use is described. We will also see how the "long tale" theory has manifested itself through Web 2.0 technology.

KEYWORDS:

Web 2.0, XML, JSON, RSS, Atom, CSS, DOM, JavaScript, AJAX, Adobe Flex, Microsoft Silverlight, security

TABLE OF CONTENTS

1 INTRODUCTION	6
1.1 Web 1.0 to Web 2.0	7
2 TECHNOLOGIES	6
2.1 XML	7
2.2 JSON	8
2.3 RSS and Atom	9
2.4 CSS	10
2.5 DOM	10
2.6 JavaScript	10
2.7 AJAX	11
2.8 Adobe Flex	18
2.9 Microsoft Silverlight	21
3 SECURITY.....	22
4 USER APPLICATIONS OF WEB 2.0	24
4.1 Blogs	24
4.2 Wikis	24
4.3 Tagging and social bookmarking	25
4.4 Mashup and API	26
4.5 Multimedia sharing	27
5 THIRD PARTY APPLICATIONS OF WEB 2.0	29
5.1 Business applications	29
5.2 The Long Tail theory and its application in Web 2.0	30
5.2.1 Google Adsence	32
5.2.2 Google Analytics	32
5.3 Political applications	32
6 CONCLUSION.....	35
REFERENCES	37

PICTURES / FIGURES

Figure 1. The traditional model for web applications (left) compared to the Ajax model (right) (Garrett J. J., 2005).	12
Figure 2. The synchronous interaction pattern of a traditional web application (top) compared to the asynchronous pattern of an Ajax application (bottom) (Garrett J. J., 2005).	14
Figure 3. During compilation, MXML code is translated into ActionScript code and then all the ActionScript codes are compiled into binary SWF files. The SWF file can be uploaded to the web server, where it is then served up upon user's request (Adobe Systems Inc. 2008).	19
Figure 4. Silverlight 2.0 architecture	21
Figure 5 . Example of tagging of web 2.0 (http://www.tagclouds.com/)	26
Figure 6. Depicts schematic representation of web 2.0 services connected by mashups (The structural evolution of the Web 2.0 service network, Junseok Hwang, Jörn Altmann, Kibae Kim, 2009).	27
Figure 7. The Long Tail (Starak, 2006)	31
Figure 8. Participation of users in the internet. (Osimo, 2008).	33

TABLES

Table 1. Evolution Web 1.0 to Web 2.0	9
---	---

1 Introduction

Web 2.0 emerged as a consequence of various combinations of web innovations during past years. Web 2.0 is associated with applications and services in the web that are interactive and user-concentrated (blogs, video sharing, social networking and podcasting). The term of Web 2.0 was first introduced by Tim O'Reilly (O'Reilly 2005). Later, Eijkman introduced more detailed definition of Web 2.0 as a trend of Internet services that promote “users to collaboratively create, share and recreate knowledge from multiple sources, leverage collective intelligence and organized action” (Eijkman, 2008). Fundamental advantage of Web 2.0 is that it allows users to contribute to the web as much as they consume the web itself (Hwang et al, 2009; Anderson P., 2007).

Technologically, there were not major breakthrough in Web 2.0 development. Web 2.0 services are based mainly on technologies and open standards that were used since the web was founded. These technologies were improved and advanced leading to formation of Web 2.0. However, Web 2.0 uses web as a starting point rather than using desktop as a technology platform (Anderson P., 2007).

This thesis covers mainly technologies and applications of Web 2.0. First chapter of this thesis presents introduction to the topic as well as evolution of Web 2.0 from Web 1.0. Second chapter presents technologies that are applied by Web 2.0 including XML (eXtensible Markup Language) that is a text format used for data storage and data exchange and its competitor JSON (JavaScript Object Notation) – another text format that is responsible for data interchange. Followed by introducing RSS (Really Simple Syndication) and Atom – attributes responsible for aggregation and syndication of data that belong to a XML based family and produce web feed formats; CSS (Cascading Style Sheets) are text files consisting of code that tell browsers how to display an HTML page and DOM (Document Object Model) that helps applications and scripts to access and edit style sheets. Finally, JavaScript is presented as a main component of Web 2.0 page. Namely, JavaScript hold everything together. Next technologies that are described in this thesis are responsible

mainly for interface of web applications: AJAX (Asynchronous JavaScript) Adobe Flex and Silverlight, these applications compete with each other. Chapter 3 presents security issues describing vulnerable parts that are mainly Cross Site Scripting (XSS), authentication and anti-automation. Chapter 4 presents applications of Web 2.0 that includes firstly, user applications and secondly, third-party applications. User applications are described as follows: blogs, wikis, tagging and social bookmarking and multimedia sharing; including descriptions of mash-ups and API (Application Programming Interface). Furthermore, Web 2.0 applications are extended to third-party applications that are business and political applications enclosed with the Long tail theory and participation of users in the internet respectively. Finally, chapter 5 presents conclusions of this work, outlining importance of Web 2.0 and suggesting those dimensions that need development.

1.1 Web 1.0 to Web 2.0

It is still questioned where the boundary between Web 1.0 and Web 2.0 lies. Is Web 2.0 expected product of technological development or is that a phenomenon by itself. Besides, it is problematic to categorize sites to Web 1.0 or Web 2.0. However, the difference between the two sites, their structure and layout, is obvious when viewed by a user. Web 2.0, in comparison to Web 1.0, is much better oriented for social, political, and business users and provides better appliances and services to such those groups of users (Cormode G. & Krishnamurthy B., 2008).

Studies show that Web 1.0 sites have hierarchical structure that consists of a front page leading to subpages with cross-links and search functions. Due to the fact that users of Web 1.0 can only view the site without interaction, Web 1.0 does not frequently require users to log on. Simplicity of Web 1.0 site structure makes them easy to study. Subsequently, Web 2.0 is more difficult to study in detail, since every site has more complicated structure of its own. Social networks for example are very user-centric and log-on is necessary. This leads to an individual user of a site to have a unique experiences on the site(Cormode G. & Krishnamurthy B., 2008). In other words, if for example a Facebook user logs in to his personal account he/she will have an individually tailored content at his/her disposal. Web 1.0 on the other hand does not allow

users to modify or add site content. YouTube is another good example of web 2.0 appliance as the site recognizes the viewed content and recommends future content accordingly.

There is significant difference between how often the Web 1.0 and Web 2.0 sites are updated. Because of the increased interactivity, Web 2.0 sites are updated more frequently. Sites are not necessarily modified by users only; comments and other types of adjustments can be done also by other users; this counts as an update as well. Web 1.0 sites on the other hand are generally updated only by its owners. Commercial sites are expected to be updated often with same time intervals and individual sites are updated randomly. Furthermore, the rate of updates is correlated to the popularity of the site.

There is difference between purpose of links between Web 1.0 and Web 2.0 sites. Web 1.0 sites have frequently links to other sites to enlarge the scope of a site. Web 1.0 site covers only specific information, whereas Web 2.0 offers individual activities and links inside the site, although log on is required. Log on is beneficial for two reasons for companies that own sites. Firstly, users can be traced easily once they have their own account. This issue, however, gives rise to security concerns. Secondly, depending of the content of an account, the site can offer individual content. Additionally, personalized advertising comes along with logging on.

Both Web 1.0 and Web 2.0 tried 'portalization' which means adding versatile options to attract users. There were attempts to create Web 1.0 portals with features like news, sports, weather etc. However, Web 2.0 provides more individual oriented portal. For instance in Web 2.0 sites it is possible for a user to add his/her own mash ups, which in practice can be personal information, images, videos, posts etc.

O'Reilly (2005) outlines Web 1.0 and Web 2.0 differences and evolution of web functions as listed in table 1. Currently Web 2.0 technologies are more used than those of Web 1.0.

Table 1. Evolution Web 1.0 to Web 2.0

Web 1.0		Web 2.0
DoubleClick	-->	Google AdSense and AdWords
Ofoto	-->	Flickr
Akamai	-->	BitTorrent
mp3.com	-->	Napster
Britannica Online	-->	Wikipedia
personal websites	-->	blogging
evite	-->	upcoming.org and EVDB
domain name speculation	-->	search engine optimization
page views	-->	cost per click
screen scraping	-->	web services
publishing	-->	participation
content management systems	-->	wikis
directories (taxonomy)	-->	tagging ("folksonomy")
stickiness	-->	syndication

2 Technologies

A Web 2.0 website may typically feature a number of the following technologies:

- AJAX (Asynchronous JavaScript and XML)
- Adobe Flex
- Microsoft Silverlight
- Syndication and aggregation of data in RSS and Atom
- Accurate and comprehensible XHTML markup and the use of Microformats
- Clean and meaningful URLs
- Broad use of folksonomies (in the form of tags or tagclouds, for example)
- Use of wiki software either completely or partially (where partial use may grow to become the complete platform for the site)
- Implementation of Open Source applications and techniques, such as the LAMP solution stack
- Weblog publishing
- Mashups
- REST or XML Webservice APIs

(O'Reilly, 2005; Adobe Inc, 2010; Microsoft, 2010; Wikipedia, 2010)

Web 2.0 experienced rapid development due to emergence of a new generation of Web-related technologies and standards. With introduction and quick spread of Web 2.0 technology the idea of the 'Web as a platform', introduced back in 1990s, seems to becoming true.

Advantages brought by Web 2.0 technologies are: applications are available to users and customers over the Internet and there is no need for installation or updates of these applications. Users will always have newest version with the most recent data at their disposal. Performance of one's hardware is not of a great importance, since all heavy calculations are handled on a server side. Nowadays, most of web services are platform independent and Web 2.0 applications are available from any machine which have a web browser and

access to Internet. Important data is accessible almost from any part of the world.

For users to be able to continue interacting with a web page, communications such as data requests going to the server are separated from data coming back to the page asynchronously. If this was not to be possible, user would have to wait for the data to come back before they can do anything else on that page. This also increases overall performance of the site, as the sending of requests can be completed faster and with greater independence with no queuing, previously required to send data back, or potential blocking.

2.1 XML

XML (eXtensible Markup Language) is derived from SGML (Standard Generalized Markup Language) and it is used for data storage and data exchange. It is not a programming language, but a simple and flexible text format which is used inside other programming languages (Jacobs, 2006). Most common example of use of XML text format is XHTML file. Structure of XML is perfect for use in web applications - the determination of structure and semantics is flexible and data is saved in simple format so it could be easily used for data exchange (Nussbaumer and Gaedke, 2006).

Example of XML file, which shows how could DVD collection be data arranged in XML file.

```
<!-- Edited by XMLSpy -->
-
<CATALOG>
-
<CD>
<TITLE>Empire Burlesque</TITLE>
<ARTIST>Bob Dylan</ARTIST>
<COUNTRY>USA</COUNTRY>
<COMPANY>Columbia</COMPANY>
<PRICE>10.90</PRICE>
<YEAR>1985</YEAR>
</CD>
-
<CD>
<TITLE>Hide your heart</TITLE>
<ARTIST>Bonnie Tyler</ARTIST>
<COUNTRY>UK</COUNTRY>
<COMPANY>CBS Records</COMPANY>
<PRICE>9.90</PRICE>
<YEAR>1988</YEAR>
</CD>
-
<CD>
<TITLE>Greatest Hits</TITLE>
<ARTIST>Dolly Parton</ARTIST>
<COUNTRY>USA</COUNTRY>
<COMPANY>RCA</COMPANY>
<PRICE>9.90</PRICE>
<YEAR>1982</YEAR>
</CD>
</CATALOG>
```

2.2 JSON

JSON (JavaScript Object Notation) is a data-interchange format competing with XML. It is based on a subset of the JavaScript Programming Language, Standard ECMA-262 3rd Edition - December 1999. JSON is a text format that is completely language independent but uses conventions that are familiar to programmers of the C-family of languages, including C, C++, C#, Java, JavaScript, Perl, Python, and many others (json.org, 2010). JSON is not associated with any kind of other data transportation pattern. When one's data comes back from the server, it is already in a JavaScript object format. In comparison, XMLHttpRequest object must be used to get XML from the server. Many developers consider JSON easier to read than XML. On the other hand it

is quite hard to compare those technologies, since one is data format (JSON) and the other a data transportation system (XML). Nevertheless, it is possible to embed JSON code into XML in order to gain benefits from both of those technologies (Markham, Daniel B., 2008).

2.3 RSS and Atom

RSS (Really Simple Syndication) is a XML based family of web feed formats used to publish frequently updated works - such as blog entries, news headlines, audio and video - in a standardized format. For a variety of historical reasons there are a number of RSS formats (RSS 0.91, RSS 0.92, RSS 1.0, RSS 2.0) and there are some issues of incompatibility. For example RSS 2.0 is not an updated version of RSS 1.0 but of a different standard altogether (W3C, 2001, RSS 1.0; W3C, 2002, RSS 2.0).

In 2003, a new syndication system called Atom was proposed and developed in order to clear up some of the inconsistencies between RSS versions and the problems with the way they interoperate. Atom consists of two standards: XML based web feed standard for syndication of data and Atom Publishing Protocol (APP), a HTTP-based protocol for creating and updating blogs, Wikis, Weblogs and other related content (IETF, 2005, RFC 4287; IETF, 2010).

Atom has many similarities with RSS and has some advantages over RSS, yet it is up to developer to decide which format to use. Google's GData and Microsoft's Project Astoria are great examples of Atom 1.0's strengths as a format, even though Yahoo for example still relies on RSS 2.0. For users it is more important that data could be presented in readable form, than in which technology it is implemented. There are various aggregator softwares and plugins available, which are compatible with most versions of RSS and Atom (Create RSS Feed, 2006; Vishal Sood, 2008, The world of Syndication: Atom 1.0 vs. RSS 2.0?).

2.4 CSS

CSS is an abbreviation for Cascading Style Sheets. Style sheets are simply text files with extension .css , composed of lines of code that tells browsers how to display an HTML page. They give the designer more control over the appearance of a webpage by allowing to specifically define styles for elements, such as fonts of the page. By using CSS one could separate HTML content from its appearance, distinguishing style from structure. There is no need to define fonts and styles on every page separately, since you can refer to just to one CSS file. It also reduce traffic, since it is enough for a browser to download into cache CSS file (Nguyen, 2003). Some of developers mention that the most important features of CSS is the ability to arrange elements in a web page - place elements and arrange text in relation to graphic elements (Crane 2006). There is tight relation between CSS and Ajax. Users can change style elements on the fly, with help of JavaScript via DOM and effecting CSS. Even inside CSS user can define events which invoke changes into design of a web page.

2.5 DOM

DOM (Document Object Model) is an interface independent of operating systems and programming languages. With its assistance applications and scripts have access to edit structure and styles of documents dynamically (W3C,2005). On the client side in web browsers DOM is used by to edit web pages (HTML, php, asp) and for reading, rearranging, structuring and creating of new XML files. On the server side, DOM is used for creation of XML files to be sent to web browser and for reading of XML files received from other sources (Darie et al., 2006).

2.6 JavaScript

JavaScript has many names; Mocha, LiveScript, JScript, ECMAScript and is one of the most popular object oriented programming languages. Mostly used for scripting in network environment. Source code of JavaScript, as an interpreted language, must be implemented and executed by JavaScript

engine. Among developers JavaScript has a bad reputation is also often referred to as world's most misunderstood programming language. There no exact standard regarding JavaScript. Disadvantage of JavaScript is in differences of implementation in browsers of different developers (Douglas Crockford, 2001).

Lack of proper design environment effected JavaScript popularity amongst developers. For a longer time, web designers had to use popup window of alert window as a debugging tool. However, nowadays there are more tools for different browsers. One of those is Firebug, a plug-in for Firefox, especially helpful in web programming.

The most important features of JavaScript are possibility to dynamically add into web pages content and functionality. JavaScript has access to edit HTML document through DOM interface. Some of JavaScript's provided features are; user can immediate interact with web page, display of small data components and displaying of their interfaces, navigation through multi-tables and plug-ins or control over Java applets, processing of user's entrees before sending them back to server, contents and styles on the fly editing of web pages corresponding to users' action (Goodman ja Morrison, 2004).

2.7 AJAX

AJAX is one of the key components of Web 2.0 architecture, abbreviation of which consist of some good old technologies – Asymmetric JavaScript and XML. The name was invented and presented by the director of Adaptive Path company, Jesse James Garrett in "Ajax: A New Approach to Web Applications"- article 2005 (Garrett J. J., 2005).

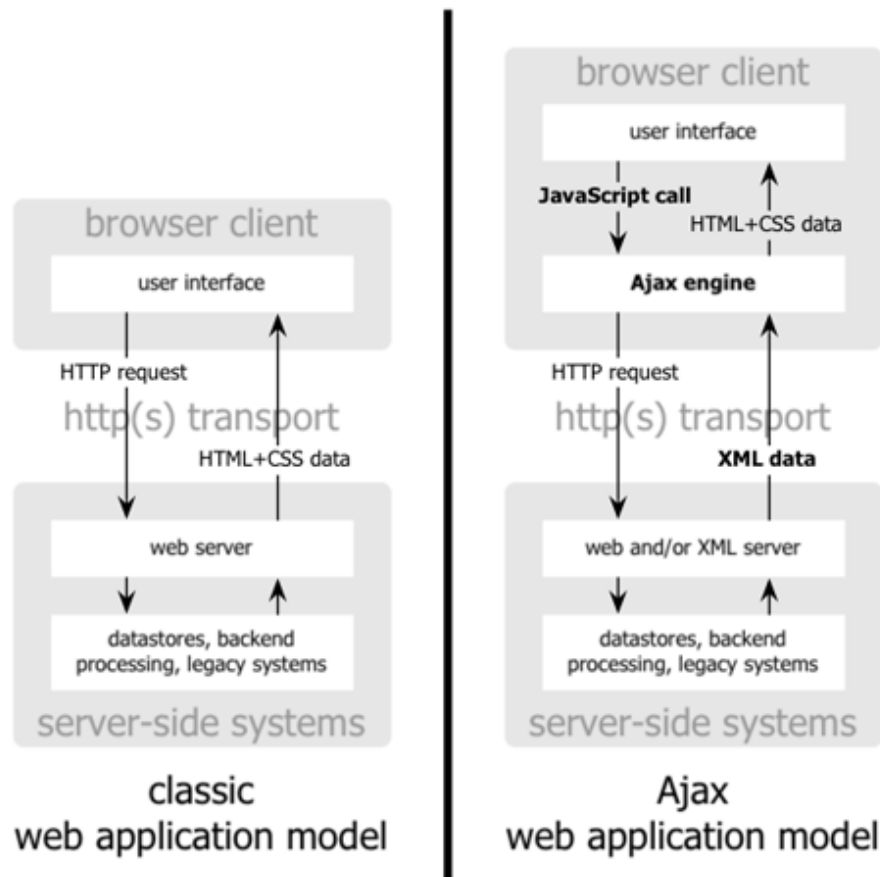


Figure 1. The traditional model for web applications (left) compared to the Ajax model (right) (Garrett J. J., 2005).

AJAX became really quickly one of the most common tools to create interface for web applications. Garrett (2005) defines Ajax as a set of independent technologies, which are really efficient together. AJAX consists of HTML and CSS which define appearance of the web page. Data exchange and data manipulation requests are formatted in XML and XSLT (Extensible Stylesheet Language Transformations) or JSON. When AJAX notices changes made by a user, it makes JavaScript use DOM to dynamically update page in accordance to the most recent information. With the XMLHttpRequest object, users can retrieve and submit XML data directly to a Web server without reloading the page. All of these components are 'glued' together by JavaScript (W3C, Scripting and AJAX, 2010; OpenAjax Alliance, Introducing Ajax and OpenAjax, 2010).

JavaScript is a main component in AJAX application. It is holding together all other components. When AJAX application is running on a client machine it participates with dataflow, appearance and logic of a webpage. JavaScript is a tool that AJAX uses to implement logic of a web page through an API (Application Programming Interface) with defined functions and object properties (Crane et al., 2006).

AJAX is used in automatically updated pages, tool hints, automatic form fillers, advanced interface components and spell check on the fly (Aselsson & Schutta, 2007). AJAX by itself does not provide methods for manipulating images or sound streaming. However, it can be implemented together with Flash technology. MacVittien (2006) states, that 93 percent of browsers in usage are AJAX compatible. Remaining seven percent are mostly users in which browser JavaScript is disabled. Numbers of AJAX users have been rapidly rising (Nielsen, 2005).

AJAX makes it possible to create new types of web applications, which have more in common with desktop applications. Since there are quite a lot of developers familiar with JavaScript and some other old technologies used in it, it is convenient and easy for them to use AJAX. Platform independence (write once, run via Web) and compatibility with all browsers makes AJAX truly appealing (Paulson, 2005).

Yet, AJAX is not only collection of benefits for developers, as it also has its weaknesses. Adding small components to AJAX applications is quite simple, but larger parts requires bigger effort (Paulson, 2005). MacVittie (2006) argues that AJAX is still an unfinished product, as compatibility between AJAX applications are close to none. According to MacVittie, biggest risks are AJAX's security concerns as well as its control and usability. Correctness of AJAX applications' data flow should be an issue for consideration and be carefully examined, so as not to undermine user's security (eg, SQL injection, cookie poisoning).

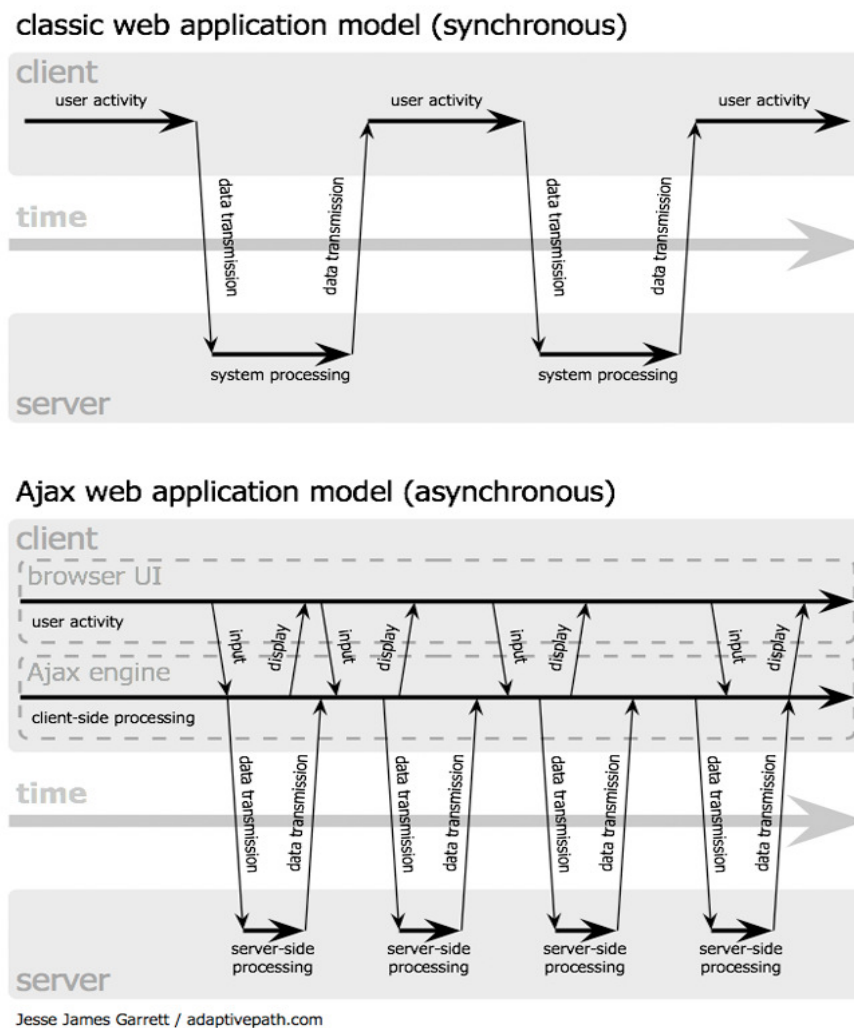


Figure 2. The synchronous interaction pattern of a traditional web application (top) compared to the asynchronous pattern of an Ajax application (bottom) (Garrett J. J., 2005).

Asleson & Schutta (2005) are reminding that in AJAX applications validation of data should be done on a server's side, since users could disable JavaScript on a browser. For developers moving to AJAX, painful memories of DHTML applications could arise. Sometimes lack of standardization in JavaScript makes things more difficult. And differences in implementations for JavaScript from browser developers could bring about some issues relating to "memory overflow".

AJAX is competing with Adobe Flash and in some part with Microsoft ActiveX and Sun Java. Weaknesses of competitors are mostly their dependency on a

platform and version. On the other hand, there are a lot of applications on the web where Flash and AJAX have been combined successfully even in commercial use. Examples of these are Flickr and Google's Streetside View. Adobe also trusts in interoperability of these technologies (Domenig, Marc, 2005).

Popularity of AJAX is rising exponentially and gives rise for the need to create new development tools. Some development tools already exist and are being updated frequently. Some of the tools are used in AJAX development GWT, YUI, Dojo, Spry, Atlas, xAjax. Since decent development tools are available for AJAX, even more complex, reliable and important applications could be developed with less effort.

One of a crucial concern in the future of AJAX is standardization. Even though separate components of AJAX are standardized (EcmaScript, DOM), AJAX as a whole remains quite poorly standardized. OpenAjax Alliance (openajax.org) has been created in 2006 to promote openness and compatibility of AJAX technologies (OpenAjax Alliance, 2010).

Example below shows how a web page helps user with name suggestion hints, while user types characters. It is a demonstration of how a web page could communicate with a web server without reloading whole page.

```
<html>
<head>
<script type="text/javascript">
function showHint(str)
{
if (str.length==0)
{
document.getElementById("txtHint").innerHTML="";
return;
}
if (window.XMLHttpRequest)
{// code for IE7+, Firefox, Chrome, Opera, Safari
xmlhttp=new XMLHttpRequest();
}
else
{// code for IE6, IE5
xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
}
xmlhttp.onreadystatechange=function()
{
if (xmlhttp.readyState==4 && xmlhttp.status==200)
```

```

        {
            document.getElementById("txtHint").innerHTML=xmlhttp.responseText;
        }
    }
xmlhttp.open("GET","gethint.php?q="+str,true);
xmlhttp.send();
}
</script>
</head>
<body>

<h3>Start typing a name in the input field below:</h3>
<form action="">
First name: <input type="text" id="txt1"
onkeyup="showHint(this.value)" />
</form>
<p>Suggestions: <span id="txtHint"></span></p>

</body>
</html>

```

(W3Schools, 2010, AJAX PHP Example)

Source code explanation: When a user types a character in the input field above, the function "showHint()" is executed. The function is triggered by the "onkeyup" event. If the input field is empty (str.length==0), the function clears the content of the txtHint placeholder and exits the function.

If the input field is not empty, the showHint() function executes the following:

- Create an XMLHttpRequest object
- Create the function to be executed when the server response is ready
- Send the request off to a file on the server
- Notice that a parameter (q) is added to the URL (with the content of the input field)

The source code in "gethint.php" below, checks an array of names, and returns the corresponding name(s) to the browser:

```
<?php
// Fill up array with names
$a[]="Anna";
$a[]="Brittany";
$a[]="Cinderella";
$a[]="Diana";
$a[]="Eva";
$a[]="Fiona";
$a[]="Gunda";
$a[]="Hege";
$a[]="Inga";
$a[]="Johanna";
$a[]="Kitty";
$a[]="Linda";
$a[]="Nina";
$a[]="Ophelia";
$a[]="Petunia";
$a[]="Amanda";
$a[]="Raquel";
$a[]="Cindy";
$a[]="Doris";
$a[]="Eve";
$a[]="Evita";
$a[]="Sunniva";
$a[]="Tove";
$a[]="Unni";
$a[]="Violet";
$a[]="Liza";
$a[]="Elizabeth";
$a[]="Ellen";
$a[]="Wenche";
$a[]="Vicky";

//get the q parameter from URL
$q=$_GET["q"];

//lookup all hints from array if length of q>0
if (strlen($q) > 0)
{
    $hint="";
    for($i=0; $i<count($a); $i++)
    {
        if (strtolower($q)==strtolower(substr($a[$i],0,strlen($q)))
        {
            if ($hint=="")
            {
                $hint=$a[$i];
            }
            else
            {
                $hint=$hint." , ".$a[$i];
            }
        }
    }
}
```

```
// Set output to "no suggestion" if no hint were found
// or to the correct values
if ($hint == "")
{
    $response="no suggestion";
}
else
{
    $response=$hint;
}
```

```
//output the response
```

```
echo $response;
```

```
?>
```

(W3Schools, 2010, AJAX PHP Example)

2.8 Adobe Flex

Adobe Flex is another technology often used in Web 2.0 applications. Flex was developed by Adobe in 2006 to create Rich Internet Applications (RIA). A major advantage of Flex is its ability to play audio and video files. Good examples are YouTube and SoundCloud. Flex made it possible to integrate multimedia and interactive content into web pages with standard HTML.

Flex applications could be created by MXML and ActionScript programming languages. Flex applications are presented by a Flash Player which only understands ActionScript. Meaning that any of MXML code should be transformed by compiler to Action Script which in its turn is compiled to SWF binary.

Flash Player and Flash Player plug-in for browser are used for displaying SWF files. For developers there are tools available, such as Flex builder or Eclipse with a Flex plug-in (Corlan Mihai, 2009).

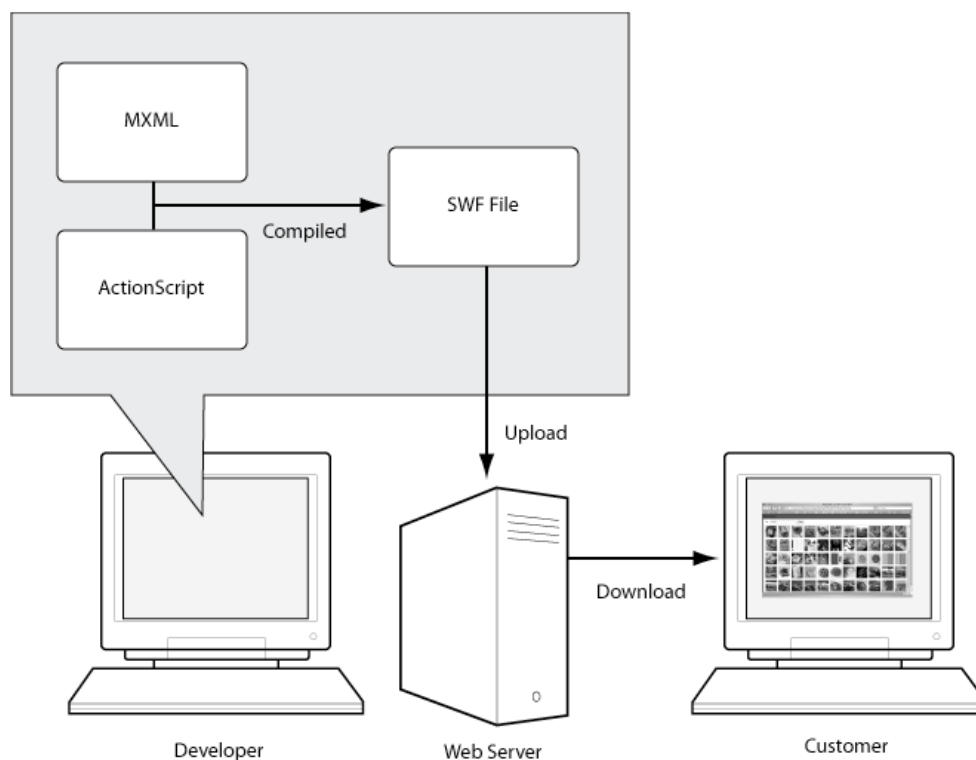


Figure 3. During compilation, MXML code is translated into ActionScript code and then all the ActionScript codes are compiled into binary SWF files. The SWF file can be uploaded to the web server, where it is then served up upon user's request (Adobe Systems Inc. 2008).

Principle of Flex functioning is different in comparison to legacy web page. Server is sending to a client compiled by Flex application SWF file, which is displayed by Flash Player. It is quite common that SWF file contains only functional algorithms of Flex application on a client side. Additional data content which Flex application requests on demand from server's database makes it possible to create lighter SWF files with shorter loading times, since not all of the content is needed by a user. All data provided by server is in XML, JSON or AMF3 format. Client-side Flex application is responsible for displaying all this information for a user. In this case we have service oriented architecture: Flex application is acting as a client, which could use communication services. Flex application can change its state without reloading webpage or SWF file in browser. Flex application is a client which can do more than just display data from server. By using Flex and Flash Player it is possible to create almost any

Internet application, from games and widgets to complex web applications (Corlan Mihai, 2009).

Example below shows how to upload file to the server and get confirmation if the file has been correctly processed.

```
<?xml version="1.0" encoding="utf-8"?>
<mx:Application xmlns:mx=http://www.adobe.com/2006/mxml
  layout="vertical"
  verticalAlign="middle"
  backgroundColor="white"
  creationComplete="init();" >
  <mx:Script>
    <![CDATA[
      import mx.controls.Alert;
      private var fileRef:FileReference;
      private var urlReq:URLRequest;
      private function init():void {
        fileRef = new FileReference();
        fileRef.addEventListener(Event.SELECT, fileRef_select);
        fileRef.addEventListener(Event.COMPLETE, fileRef_complete);
        fileRef.addEventListener(IOErrorEvent.IO_ERROR,
fileRef_ioError);
        fileRef.addEventListener(DataEvent.UPLOAD_COMPLETE_DATA,
fileRef_uploadCompleteData);
        urlReq = new URLRequest();
        urlReq.url = "http://localhost/uploader.php";
      }
      private function fileRef_uploadCompleteData(evt:DataEvent):void
      {
        //Data from server
        trace(evt.data)
      }
      private function start():void {
        fileRef.browse();
      }
      private function fileRef_select(evt:Event):void {
        fileRef.upload(urlReq);
      }
      private function fileRef_complete(evt:Event):void {
        Alert.show(evt.toString(), evt.type);
      }
      private function fileRef_ioError(evt:IOErrorEvent):void {
        Alert.show(evt.text, evt.type);
      }
    ]]>
    </mx:Script>
    <mx:Button label="upload" click="Let's do it();" />
  </mx:Application>
```

(Flex Cookbook, 2008)

2.9 Microsoft Silverlight

Microsoft Silverlight is a platform and browser independent plug-in, which makes possible to run web applications consisting of animations, vector graphics and multimedia files. In Silverlight applications, user interfaces are declared in Extensible Application Markup Language (XAML) and programmed using a subset of the .NET Framework. XAML can be used for marking up the vector graphics and animations. Competing with AJAX, Silverlight makes it possible to access XML data through a DOM interface for further interaction. (Microsoft, 2010; Wikipedia 2010).

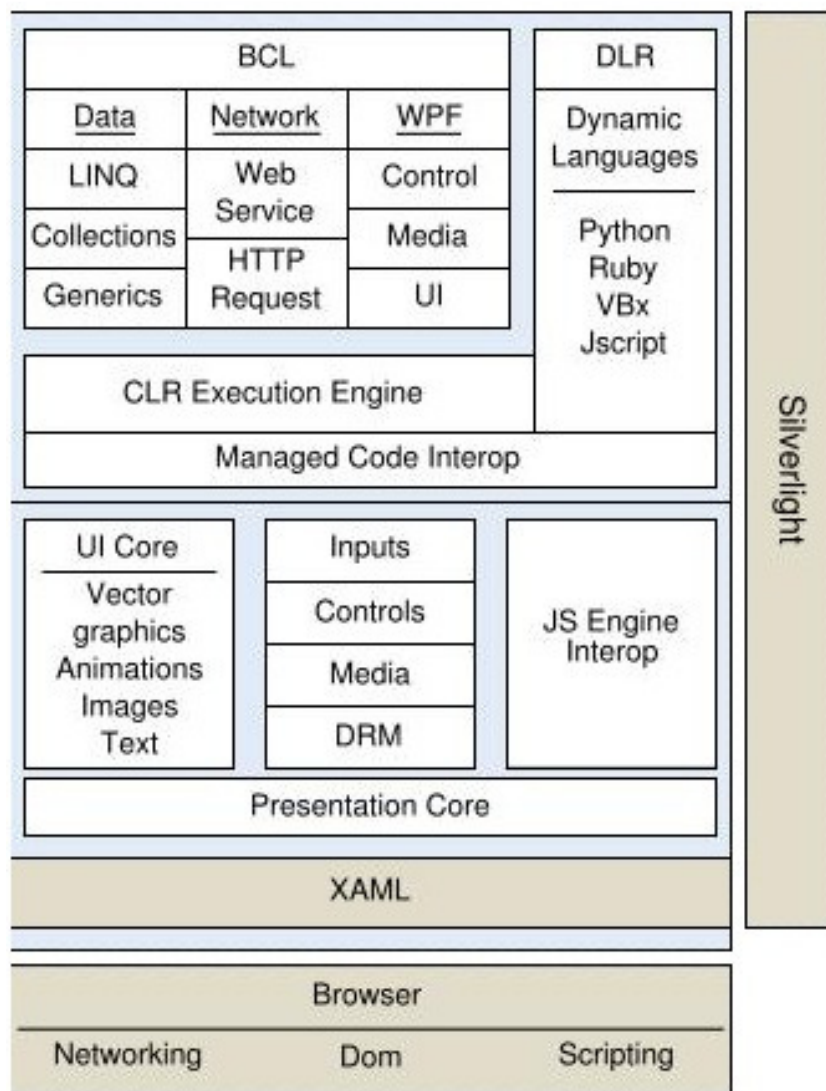


Figure 4. Silverlight 2.0 architecture

3 Security

Cross Site Scripting (XSS) remains at the top of the list of vulnerabilities plaguing Web 2.0 websites (WhiteHat Security Inc, 2008). One of XSS attack scenarios could be stored XSS vulnerability, malicious input sent by a hacker is stored in the system then displayed to other users. Systems that allow users to use HTML tags or other formatted content are most vulnerable to attacks. At risk are blogs, social networks, and wikis (OWASP, 2010). Cross Site Request Forgery (CSRF) attacks could happen when victim visits what appears to be an innocent-looking web site, but which contains malicious code which generates requests to a different site instead. Due to popularity of Web 2.0 technologies, which allow applications to communicate without visual feedback, it is easier for hackers to use CSRF. Since in legacy web applications, most user-generated requests produced a visual effect on the screen, making CSRF easier to notice (OWASP, 2010).

Authentication control is quite crucial when company stores sensitive data online. In many Web 2.0 applications, access is gained to numerous users, not just qualified personnel. It makes whole system vulnerable, because there is a chance that a less experienced user will make a change that will negatively affect the overall system. One of such examples could be mistaken edit on Wikipedia which is then accepted as fact by many of the site's visitors. Inexperienced users and purely adjusted authentication control settings could leave system open for hackers' exploits. Which could gain access to a greater number of administrative accounts. Also passwords can often be easily cracked if the incorrect security controls are enabled. The systems may have insufficient brute-force controls, permit clear text passwords, or have been tied together in a single-sign-on environment, making an attack easier. (Morrison, Rick 2007) Phishing is one of the threats in modern web space. The risk to get on a fake web site instead of genuine is great. Just misspelled URL address and user could end up on a malicious web page, where they are asked to update personal information, such as passwords and credit card, social security or bank account numbers (OWASP, 2010).

Since the Web 2.0 technologies became popular, new injection attacks appeared. Such as XML injection, XPath injection, JavaScript injection, and JSON injection. In addition, because Web 2.0 apps often rely on client side code, they more often perform some client-side input validation which an attacker can bypass (OWASP, 2010).

Insufficient Anti-automation takes place when a web application permits to automate a process that should only be performed manually by a human. Web 2.0 technology let hackers automate attacks easier. An automated robot (application) could execute thousands of requests a minute, causing loss of performance or service failure. It also could be used for opening of thousands of accounts. Anti-automation mechanisms like Captchas have been presented to stop or at least slow down these types of attacks (The Web Application Security Consortium, 2010).

4 User applications of Web 2.0

Web 2.0 user applications are blogs, wikis, multimedia sharing, syndication, podcasting and tagging services. The technology underlying these services are relatively simple. However, using various combinations of these technological blocks and open standards enables building well working web 2.0 sites. Web 2.0 develops over time increasing features of sites. Some of the sites are well developed and used while some still undergo changes (Anderson P., 2007).

4.1 Blogs

Definition of blog was introduced in 1997 by Jorn Barger. The web, according to Barger is a webpage that consists of paragraphs of personal diary entries, opinion, information and posts. These entries are posted in chronological order. Other users can comment on entries in most blogs. This enables conversation between a blog holder and a readers.

Bloggers use tagging option to give their posts keyword/keywords; readers can use these tags to read posts of authors that have the same tag. Linking is possible also between blogs and commonly used to indicate the source that blogger used (Anderson P., 2007).

4.2 Wikis

Wiki is a service that allows users to edit content of a web page if user has an access to this web site. Links are used in wikis to crosslink pages inside the site and outside of it (Leuf & Cunningham, 2010). Wiki pages can be edited by other users, the content of the page can be edited and even deleted. However original versions are stored and can be revised by service providers.

Simplicity and openness of wikis explain the success of for example Wikipedia. However, openness causes problems associated with vandalism and low quality inputs. However, wikis can be used in educational purposes

as well. Students can for example collaborate on the same project from wikis (IIB, 2010).

4.3 Tagging and social bookmarking

Social bookmarking is emerging user-targeted feature on the web; social bookmarking means adding a reference link (rather than adding a file) to a social network site. Tagging evolves in folksonomy (user-generated tags) and taxonomy (collaborative official tagging) (Golder & Huberman, 2006). Del.icio.us is an example of tagging; one can add a tag to own profile and thereby share own opinion with others and simultaneously contribute to a topic. Another example of tagging is CiteULike - cite that is used for academic purposes. The metadata of publication (title, authors, journal name etc.) are saved to user's bookmarks after user added the publication. Later these details can be used in citing the article.

Tagging cloud is an interesting setup commonly used in web. Tagging clouds represent weighted list in visual design, usually cloud is arranged in alphabetical order and words that are used more frequently are larger in size (Tag Clouds, 2010).



Figure 5 . Example of tagging of web 2.0 (<http://www.tagclouds.com/>)

4.4 Mashup and API

API stands for Application Programming Interface and is a web tool that provides feedback in forms of software applications by a request of another computer program. Basically, API is a software-to-software interface, meaning that user is not aware of functions that API is busy with. Additionally, API is using single or set of functions providing multiple developing opportunities (NPW, 2010).

A web 2.0 mashup is a Web application that integrates data retrieved from external data sources with a purpose to create new services. Mashup gets its data from a third-party; reforming that data in such a way that it gets value for users. In other words web 2.0 mashup is a Web 2.0 service composed of one or more web 2.0 services (NPW, 2010).

APIs can allow users to mash up one or more Web 2.0 services in order to create their own services (Floyd et al, 2007). Web 2.0 APIs are used by companies that own the site for users (individual modifying of site) and for

other companies (advertisement) (Hwang et al, 2009). The benefit of mashups are possibility to sustain labor sharing between providers.

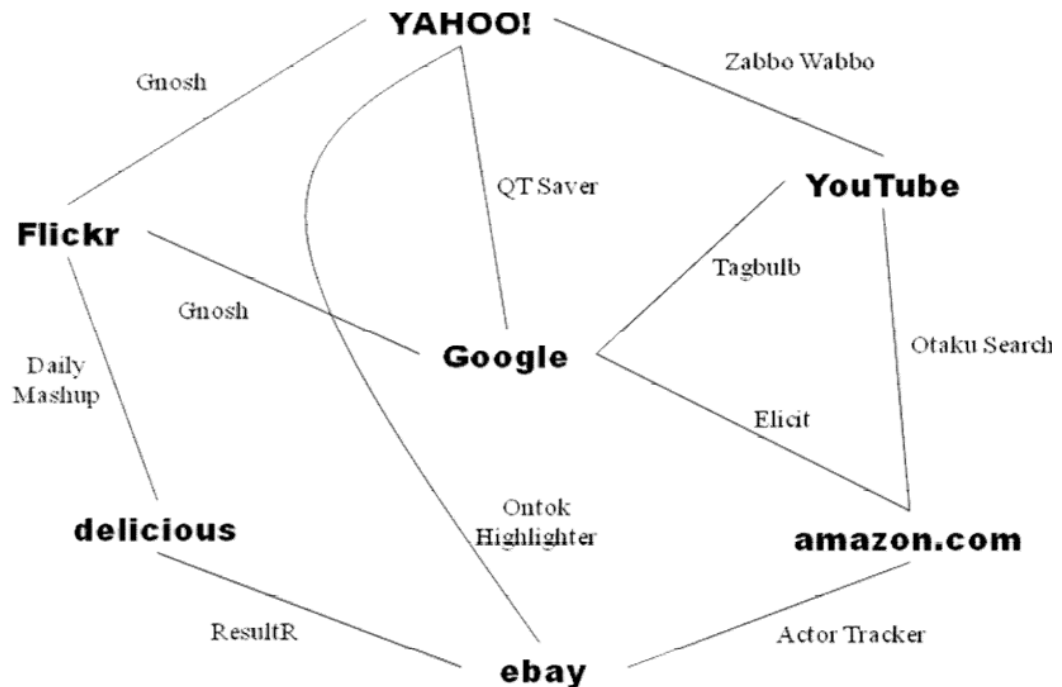


Figure 6. Depicts schematic representation of web 2.0 services connected by mashups (The structural evolution of the Web 2.0 service network, Junseok Hwang, Jörn Altmann, Kibae Kim, 2009).

4.5 Multimedia sharing

Multimedia sharing on the web gained popularity due to the development of digital media devices (cameras, videocameras, even telephones). Media sharing includes photo, video and audio sharing. Site called Flickr is a good example of photo sharing. On this site, users can upload own photographs as well as view and comment photographs of others. There is option of adding friends and attending groups on the Flickr (Breslin et al, p 122-130).

Video and audio sharing via the Internet are forms of podcasting. Word podcasting originated from concepts of iPod and broadcasting. However, audio files can be played by other audio devices as well. Users can learn about new audio files available on the internet via syndication. Furthermore,

syndication enables automatic downloading of files via RSS and Atom (Breslin et al, p 122-130).

Video podcasting works in the same way as audio podcasting. Video podcasting is known by following names as well: vlogs, vodcasting and video blogging. Digital video files are compressed before they are uploaded to the web. There are still size limitations on the web for video files (Breslin et. al., p 122-130).

5 Third party applications of Web 2.0

Third party applications are represented as business and political applications. Success of business internet protocols and applications is explained with the “long tail” theory. Participation overview is merged with political applications. Moreover, positive potential as well as risks are enclosed with political applications.

5.1 Business applications

Internet marketing develops alongside internet technologies. Business potentials of web 2.0 applications were neglected in the beginning. However, it was established that people are not attracted to conventional marketing in the same way as they used to be. Thus, Web 2.0 offers new possibilities to implicate marketing strategies. Consumers tend to trust more on independent reviews than commercials. Nevertheless, professional reviews do not have the same impact on customer’s decisions in comparison to reviews of peer customers (Constantinides & Fountain, 2008).

On the other hand, in some cases Web 2.0 had undermined the purpose of customer reviews. People started to share their experiences of products on the web with less than flattering reviews. However, there is also a good side to this development. Companies receive feedback from consumers and have potentials to improve their products in order to meet the expected requirements (Constantinides & Fountain, 2008).

In case of videos, the major drawback are copyrights. Moreover, people are often misled by content and purpose of amateur videos. There are videos with commercial content and videos with non-commercial content that still use popular products in their video material. People often don’t recognize the fine line between intended and unintended advertisement. Another dangerous issue is that videos are not pre-screened before they are viewed by others and might contain confusing and embarrassing content (Constantinides & Fountain, 2008).

Companies can benefit from web 2.0 services in following ways:

1. Informing leading bloggers etc. about new products in order to get the review for own product
2. Observe and react on the reviewers feedback
3. Gain fresh ideas from people not included in marketing industry
4. Introducing products via social networks

5.2 The Long Tail theory and its application in Web 2.0

Success of internet applications like Google, Amazon, Netflix (movies) and iTunes can be explained with the “long tail” theory (discovered by Anderson C., 2004). Figure 7 depicts the “long tail” theory; hits based on popularity in the head of short tail and the rest of bias in the long tail. The long tail theory applies to the 20 % and 80 % rule. Only 20 % of existing products are 'hits'. This rule was discovered as universal and applies to movies, music and books (Hintikka, 2007; Karch, 2008).



Figure 7. The Long Tail (Starak, 2006)

Differences of market and internet are introduced in order to understand the meaning of long tail theory. There is a problem with stores where 100 % coverage of products is not possible and only 20 % of products is available (target of marketers). This problem dissolves in the internet. Digital items fit in the digital storage very efficiently space-wise; in contrast to the lack of space in the retails (Karch, 2008).

Long tail theory underlies successful internet projects. Google, Amazon and Yahoo are able to include into their range long tail items. Covering the wide range of topics in marketing is beneficial for both consumers as well as marketers. Popularity of Google and Yahoo leads to high incomes for these companies. However, they make the most of their profit on the advertisement (Hintikka, 2007).

5.2.1 Google AdSense

AdSense is a feature of Google that can be used by websites for advertising purposes. Google serves advertisements based on search word, location and language of the user. These advertisements became popular because they are relevant to the content of a web-page. Furthermore, the owner of a website can have benefit from provisions once clicking an advertisement. Any sites can use AdSense. AdSense is a good choice for advertisement for small companies (Hintikka, 2007).

5.2.2 Google Analytics

Google Analytics is a very powerful tool in the developments of advertisements. It can help companies to target the market and to enroll more customers. Google Analytics shows specific information on the people that enter the site - number of people, language of browser, keywords etc (Google Analytics, 2010).

5.3 Political applications

Among other dimensions of Web 2.0, political applications are of a great importance. Web 2.0 is already used by politicians - citizens can participate in political discussions on the web. The issue of participation of citizens was first presented by Arnstein S. (1969). Back in 1960 citizens had different idea of how they can participate in political issues. Nowadays, we live in an era of open journalism, when every person is capable of expressing their political interests and views. Internet participation is depicted in Figure 8. The whole circle represent all users that are on the web. In Europe its is around 50 % of population (Osimo, 2008).

The core of the circle represents 3 % users that are most active; these are users that write blogs, write wikipedia articles and upload videos on Youtube. The second circle of 10 % includes those people that react on the existing content by making reviews, feedback and commenting. The next circle encloses 40 %, which are those people that use information from previous

two circle in decisions they make as consumers. The last circle completes 100 % and represents passive users that do not generate web 2.0 content but read, watch and listen what others put. These users are however reachable by web 2.0 applications.(Osimo, 2008).

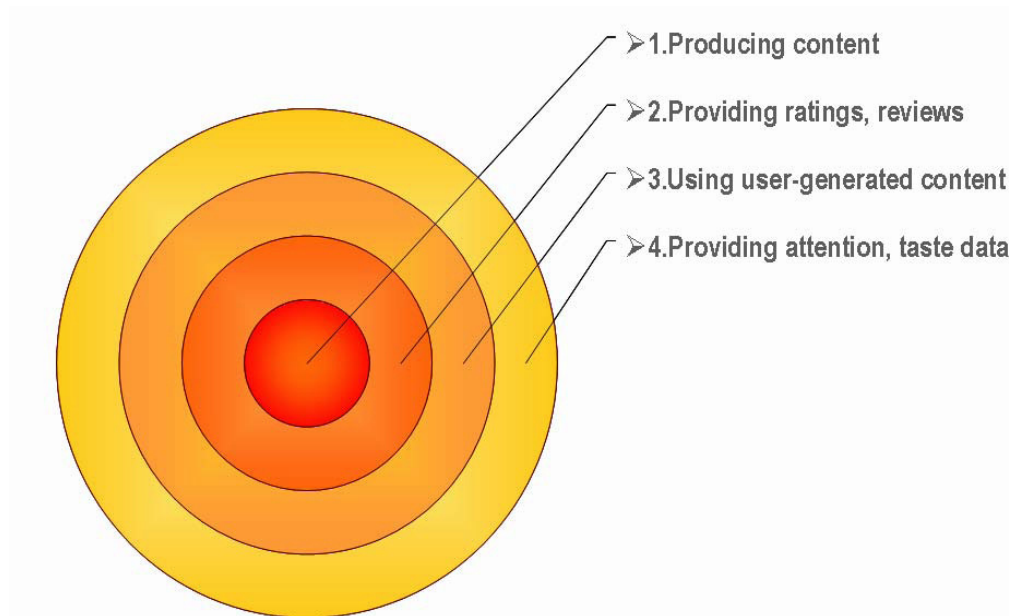


Figure 8. Participation of users in the internet. (Osimo, 2008).

Web 2.0 applications can be included in government context to enhance participation of citizens, and social servants in political, societal and environmental planning. Spatial planning is used by a public sector to enhance activities of people in territorial (local and regional) planning, ending with national and even EU level.

Another web 2.0 aspect that can be used in government context is to improve productivity of public services. The web 2.0 cannot solely improve the system - organizational rearrangement is needed as well. Major changes in private sector (fast response, higher quality of process) lead to higher expectations of citizen towards public sector services. This leads to more investment in public sector services. Citizens want to avoid irrelevant and overlapping procedures, to reduce waiting times, to reduce re-entering information (EU, 2003). Citizens expect improved productivity from public services; meaning

that citizens want faster, better and more output for given input. Meeting this requirements lead to rearranging political system. eGovernment is aiming towards more open, transparent and more accountable government; meaning that taxpayers can access information held by public sector and see the pattern how their money are distributed (EU, 2003).

Expectations of citizens to reform government does not merge with the participation of citizens in web 2.0 applications launched by government. The success of Web 2.0 entertainment and social issues did not lead to the same success on the level of governance. Osimo D. studied this problem in 2008 by case studies and discovered that web 2.0 can be employed successfully in government context as well if risk are mitigated and content made more user friendly.

Risks associated with web 2.0 applications in government context are as follows (Osimo, 2008):

- Low participation rate
- Participation restricted to elite
- Low quality of contributions
- Loss of control due to excessive transparency
- Destructive behaviour by users
- Manipulation of content by interested parties

6 Conclusion

Since the term of Web 2.0 was first introduced by Tim O'Reilly in 2005 popularity of Web 2.0 raised exponentially. As we learned technologically, there were not major breakthrough in Web 2.0 development. Web 2.0 services are based mainly on technologies and open standards that were used since the web was founded. By combining those technologies it makes possible to create interactive, exciting and impressive web pages. And one thing is certain: we are going to see more interesting and useful web applications deployed on AJAX, Adobe Flex, Microsoft Silverlight or some other competing technologies.

User applications of Web 2.0 such as blogs, wikis, multimedia sharing, syndication, podcasting and tagging services changed life of many people. The technology underlying these services are relatively simple which enables easy building of working Web 2.0 sites.

We have also acknowledged that Web 2.0 enables marketers to evaluate their products' successes and short coming with greater efficiency due to the technologies that Web 2.0 facilitates. Use of for example social networks and blogs allows customers to give valuable feedback, reviews, and product improvement suggestions to the marketers and product developers in a more spontaneous manner. All that business interests have to do is observe, react and act to meet the needs of their customers. Thus, Web 2.0 has enabled totally new type of marketing strategies since its introduction.

Web 2.0 technologies, as we noted above, has also great applicability in societal and political spheres. Exchange of information and opinions has been facilitated greatly since introduction of new Web 2.0 enabled technologies. For example, citizens can voice their opinions faster and in greater numbers and due to Web 2.0 many procedures have become less overlapping. Web 2.0 also reduces waiting times in for example administrative offices which increases the speed in which information is processes and handled.

We have also learned that Web 2.0 has also its challenges. When working with

Web 2.0 technologies, it's important to understand all kinds of risks involved. Web 2.0 may involve different types of challenges, that have not been present in legacy applications. That does not mean that the situation is getting worse, just that technology is getting more complicated. On the bright side, security measures are also becoming more advanced. Opportunities of technology provided by Web 2.0 are truly great and thus, benefits overshadow the risks.

REFERENCES

- Adobe Systems Inc., Adobe Flex, 2010, <http://www.adobe.com/fr/products/flex/>
- Adobe Systems Inc., Get oriented to Flex, 2008,
<http://learn.adobe.com/wiki/display/Flex/Get+oriented+to+Flex>
- Anderson C. (2004) *The Long Tail*. Wired. October
- Anderson P. (2007) *What is Web 2.0? Ideas, technologies and implications for education* JISC Technology and Standards Watch p.4, p 7-12
- Arnstein S. (1969) *A Ladder of Citizen Participation*. Journal of the American Institute of Planners, July
- Asleson, R. & Schutta, N. (2005). *Foundations of Ajax*. Apress, Berkeley, CA.
- Breslin J.G., Passant, A., Decker S. (2009) *The Social Semantic Web*, DO Springer-Verlag Berlin Heidelberg. 122-130
- Constantinides E., Fountain S.J. (2008) *Web 2.0: Conceptual foundations and marketing issues* .Journal of Direct, Data and Digital Marketing Practice, 9; 231–244
- Corlan Mihai, 2009, Flex for PHP Developers
<http://corlan.org/flex-for-php-developers/>
- Cormode G. & Krishnamurthy B. (2008) *Key Differences between Web1.0 and Web2.0* AT&T Labs–Research
- Crane et al., 2006, Dave Crane, Eric Pascarello, Darren James, Ajax in Action. Manning, 2006. Eijkman, 2008, p. 94
- Create RSS Feed, 2006, <http://www.create-rss-feed.com/rss-or-atom.htm>
- Crockford, Douglas, 2001, *JavaScript: The World's Most Misunderstood Programming Language*, <http://www.crockford.com/javascript/javascript.html>

Darie & al., 2006, Cristian Darie, Bogdan Brinzarea, Filip Cherecheș-Toșa and Mihai Bucica, *AJAX and PHP: Building Responsive Web Applications*. Pact Publishing, 2006.

Domenig, Marc, 2005, *Rich Internet Applications and AJAX*,
<http://www.javalobby.org/articles/ajax-ria-overview/>

Eijkman, H. (2008) *Web2.0 as a non-foundational network-centric learning space* Campus-Wide Information Systems. 25(2):93-104.

EU. Commission of the European Communities (2003) *Communication From the Commission to the Council, the European Parliament, the European Economic and Social Committee and the Committee of the Regions*

Flex Cookbook, 2008 <http://flexcookbook.ru/2008/11/filereference-class-uploadcompletedata-event-to-capture-data-from-a-server-side-script/>

Floyd, I.R., Jones, M.C., Rathi, D., Twidale, M.B. (2007) *Web mash-ups and patchwork prototyping: User-driven technological innovation with Web2.0 and open source software*. The 40th Hawaii International Conference on System science. 86

Garret, Jesse James Garrett *Ajax: A New Approach to Web Application*, 2005
<http://www.adaptivepath.com/ideas/essays/archives/000385.php>

Goodman & Morrison, 2004 Danny Goodman, Michael Morrison. *JavaScript Bible: 5th edition*. Wiley, 2004.

Golder S. & Huberman B.A. (2006) *Usage of Patterns of Collaborative Tagging Systems*. Journal of Information Science. 32(2):198-208

<http://www.google.com/analytics/>

Hintikka A.H. (2007) *Web 2.0 - johdatus internetin uusiin liiketoimintamahdollisuuksiin*. TIEKE Tietoyhteiskunnan kehittämiskeskus ry:n julkaisusarja. osa 28

Hwang J., Altman J., Kim K. (2009) *Structural Evolution of the Web 2.0 Service Network*. TEMP Discussion Paper.14

Karch M. (2008) *What Is the Long Tail and How Does It Apply to Google*

<http://google.about.com/od/googleforbusiness/f/longtailfaq.htm>

NPW (The new path web), (2010) *What is an API – Application Programming Interface*

<http://www.newpathweb.com.au/what-is-an-api-application-programming-interface>

NPW (The new path web), (2010) *What is a web mash-up*

<http://www.newpathweb.com.au/what-is-a-web-mashup>

IETF (The Internet Engineering Task Force), RFC 4287, 2005,

<http://tools.ietf.org/html/rfc4287>

IETF, 2010, Atom Publishing Format and Protocol (atompub) (concluded WG)

<http://datatracker.ietf.org/wg/atompub/charter/>

IIB (The instructional Innovations Blog), (2010) Office of Instructional Consulting, School of Education, Indiana University

<http://icoblog.wordpress.com/2010/01/22/using-wikis-to-promote-student-collaboration/>

Jacobs,2006 Sas Jacobs. *Beginning XML with DOM and Ajax*. Apress, 2006.

json.org, 2010, Introducing JSON, <http://www.json.org/>

Leuf B., Cunningham W. (2010) Wiki Org. <http://wiki.org/wiki.cgi?WhatIsWiki>

Markham, Daniel B., 2008, *JSON vs XML - A Jason vs Freddie Sequel*, AJAX & RIA Journal, <http://ajax.sys-con.com/node/203935>

Microsoft, 2010, <http://www.microsoft.com/silverlight/what-is-silverlight/>

MacVittie L. (2006). *Danger 2.0. Network Computing*. Oct 26, 40-53.

Morrison, Rick 2007, Web 2.0 Access Control

<http://www.site-reference.com/articles/Website-Development/Web-2-0-Access-Control-Part-1.html>

Nguyen, Tuyet, 2003, Wellesley College Information Services,
<http://www.wellesley.edu/Computing/Dreamweaver/CSS/cssMain.html>

Nussbaumer and Gaedke, 2006 Martin Nussbaumer, Martin Gaedke,
Technologies for Web Applications.

Nielsen, J. (2005). Why Ajax Sucks (Most of the Time). Jacob Nielsen's
Alertbox. Dec 2005. <http://www.usabilityviews.com/ajaxsucks.html>

OpenAjax Alliance, 2010, Introducing Ajax and OpenAjax,
<http://www.openajax.org/whitepapers/Introducing%20Ajax%20and%20OpenAjax.php>

O'Reilly, Tim, O'Reilly Media Inc, 2005, What Is Web 2.0,
<http://oreilly.com/web2/archive/what-is-web-20.html>

O'Reilly, T. (2007), *What is Web2.0: Design patterns and business models for the next generation of software* Communications & Strategies. 65(1):17-37.

OWASP, 2010, Cross-site Scripting (XSS)
http://www.owasp.org/index.php/Cross-site_Scripting_%28XSS%29

OWASP, 2010, Cross-Site Request Forgery (CSRF)
<http://www.owasp.org/index.php/CSRF>

Paulson, L. D. 2005, Building Rich Web Applications with Ajax. *Computer*, vol. 38, no. 10, 14-17.

OWASP, 2010, Phishing
http://www.owasp.org/index.php/Phishing#What_is_Phishing.3F

OWASP, 2010, Category:Attack
<http://www.owasp.org/index.php/Category:Attack>

Starak Y, 2006, The Long Tail, <http://www.entrepreneurs-journey.com/539/the-long-tail/>

Tag Clouds, 2010, <http://www.tagclouds.com/>

The Web Application Security Consortium, 2010, Insufficient Anti-automation, WASC Threat Classification,

<http://projects.webappsec.org/w/page/13246938/Insufficient%20Anti-automation>

Vishal Sood, 2008, The world of Syndication: Atom 1.0 vs. RSS 2.0?

<http://blogs.iis.net/vsood/archive/2008/10/06/the-world-of-syndication-atom-1-0-vs-rss-2-0.aspx>

W3C, 2005, Document Object Model (DOM) <http://www.w3.org/DOM>

W3C, 2001, RSS 1.0, <http://validator.w3.org/feed/docs/rss1.html>

W3C, 2002, RSS 2.0, <http://validator.w3.org/feed/docs/rss2.html>

W3C, Scripting and AJAX,2010,

<http://www.w3.org/standards/webdesign/script.html>

W3Schools, 2010, AJAX PHP Example,

http://www.w3schools.com/Ajax/ajax_asp.php.asp

Wikipedia, 2010, http://en.wikipedia.org/wiki/Microsoft_Silverlight

WhiteHat Security Inc, 2008, p.10

<http://www.whitehatsec.com/home/assets/presentations/PPTstats032608.pdf>