

STADIA

HELSINGIN AMMATTIKORKEAKOULU

PALJON PUHUTTU AJAX

Viestinnän koulutusohjelma
Verkkoviestinnän
suuntautumisvaihtoehto
Opinnäytetyö
23.4.2007

Pilvi Rimmanen



TIIVISTELMÄSIVU

Koulutusohjelma Viestintä		Suuntautumisvaihtoehto Verkkoviestintä	
Tekijä Pilvi Rimmanen			
Työn nimi Paljon puhuttu AJAX			
Työn ohjaaja/ohjaajat Markus Norrena			
Opinnäytetyö	Aika 23.4.2007	Numeroidut sivut + liitteiden sivut 46 + 8	
<p>TIIVISTELMÄ</p> <p>Web 2.0 viittaa toisen polven rikkaisiin verkkosovelluksiin, jotka lähestyvät käytettävyydeltään työpöydän ohjelmia. AJAX (Asynchronous JavaScript and XML) on ryhmä tekniikoita, joilla rakennetaan rikkaita verkkosovelluksia. AJAX on saanut valtavasti huomiota mullistavien ominaisuuksiensa ansiosta ja se on syrjäyttämässä tuttuja malleja verkkoympäristössä. Kaikki puhuvat AJAXista, mutta harva todella tietää mistä on kyse.</p> <p>Tämän opinnäytteen tavoitteena on selvittää mistä AJAXissa oikein on kyse, miten se on mullistanut verkkosovelluksia ja mihin suuntaan se on menossa tulevaisuudessa. Tutkimuksen kohteena olivat myös AJAXin kilpailijat, joita verrattiin tähän teoreettisella tasolla.</p> <p>Alan kirjallisuuteen nojautuvassa teknisessä selvityksessä esiteltiin AJAX-tekniikat ja niiden toimintatavat AJAX-sovelluksissa. Median esille tuomien AJAXin urauurtavien hyvien ominaisuuksien lisäksi käytiin tutkimuksessa läpi kriittisesti myös sen heikkoja puolia.</p> <p>SWOT-analyysissä selvisi, että AJAXin vahvuuksia ovat nopeus ja asynkronisuus ja heikkouksista suurimpia olivat tietoturva-aukot ja navigointimallin muutoksen mukanaan tuomat ongelmat. Analyysin tuloksena AJAXin mahdollisuuksiksi nähtiin laajempi levinneisyys ja kehittyminen sekä näiden mukanaan tuomat hyödyt. Uhkina nähtiin kilpailijat ja AJAXin suosion lasku.</p> <p>Tuloksissa selvisi myös, että kilpailijat esiintyvät kuitenkin AJAXin kanssa yhteiskäytössä, minkä vuoksi niiden vertailu kilpailijoina on hankalaa. Kilpailu on tärkeää, sillä se lisää tekniikoiden kehittymistä ja standardien muodostumista ja johtaa entistä parempiin rikkaisiin verkkosovelluksiin.</p> <p>Heikkouksistaan huolimatta AJAX on käyttöominaisuuksiltaan varteenotettava malli ja säilyttää varmasti paikkansa muiden WWW-sovellustekniikoiden joukossa.</p>			
Teos/Esitys/Produktio -			
Säilytyspaikka Taideteollisen korkeakoulun kirjasto, Aralis-kirjastokeskus			
Avainsanat AJAX, Web 2.0, RIA, XMLHttpRequest, JavaScript, verkkosovellus			



Degree Programme in Media		Specialisation New Media Design
Author Pilvi Rimmanen		
Title Lots of talk about AJAX		
Tutor(s) Markus Norrena		
Type of Work Final Project	Date 23 April, 2006	Number of pages + appendices 46 + 2
<p>Web 2.0 refers to rich internet applications (RIAs) which are approaching desktop applications. AJAX (Asynchronous JavaScript and XML) is a new way to produce RIAs. With its tremendous characters AJAX is currently replacing existing web techniques. Everybody is talking about AJAX. But is it much ado about nothing?</p> <p>Aim of this thesis is to find out what AJAX is all about, describe its revolutionary features and predict AJAX's future. AJAX's competitors are also researched on a theoretical level.</p> <p>In the technical part of the thesis techniques behind AJAX are introduced. In this part it is researched how these techniques work in AJAX applications and which are their best features. This research is not all about positive features of AJAX. It also brings up AJAX's weaknesses and looks AJAX with a critical point of view.</p> <p>The SWOT analysis revealed that AJAX's strengths are fastness and asynchronous data transmissions and its biggest weaknesses are security problems and new model of navigation. According to the analysis AJAX's possibilities are wider diffusion and possible improvements in usability and on the other hand its potential threats are competitors and declining popularity.</p> <p>During the research process it turned out that competitors of AJAX often appear in collective use with it. As a result of this it was difficult to compare competitors with each others. Competition is though important because it enhances development of web techniques and speeds up the forming of standards. It also leads to production of better web applications.</p> <p>In spite of its weaknesses AJAX is noteworthy model for doing RIAs and is expected to maintain its position among other web applications.</p>		
Work / Performance / Project -		
Place of Storage Aralis Library and Information Center, Helsinki		
Keywords AJAX, Web 2.0, RIA, XMLHttpRequest, JavaScript, web application		

SISÄLLYS

1	JOHDANTO	1
2	KOHTI RIKKAAMPAA KÄYTETTÄVYYTTÄ	3
3	AJAXIN TOIMINTA.....	4
3.1	AJAX-sovelluksen ulkoasu.....	5
3.2	Tiedon kuvaaminen	7
3.3	Datan tulkitseminen solmupuuksi	9
3.4	Toimintojen toteuttaminen.....	11
4	RISUJA JA RUUSUJA	14
4.1	AJAXin vahvuudet	15
4.1.1	Nopeus	15
4.1.2	Avoin standardi	17
4.1.3	Läpinäkyvyys	18
4.1.4	Tietojen tallennus	19
4.1.5	Samankaltaisuus työpöydän ohjelmien kanssa	20
4.2	AJAXin heikkoudet	23
4.2.1	Tietoturva.....	23
4.2.2	Navigoinnin muuttuminen	24
4.2.3	Yhteensopivuus	26
4.2.4	Muistin varaaminen	27
4.2.5	Trendikkyys.....	28
4.3	AJAXin mahdollisuudet ja uhat	29
4.3.1	Mahdollisuudet tulevaisuudessa.....	29
4.3.2	Tulevaisuuden uhat.....	31
4.4	AJAXin turvattu tulevaisuus.....	34
5	YHTEENVETO.....	37
	LÄHTEET.....	42

LIITE 1: SANASTO

LIITE 2: AJAXIN KILPAILIJAT

1 JOHDANTO

Nykyaikana verkkosovelluksista pyritään kehittämään toiminnaltaan ja käytettävyydeltään samantasoisia kuin työpöydän ohjelmista. Näitä Web 2.0 -ajan kehittyneitä sovelluksia kutsutaan rikkaiksi WWW-sovelluksiksi (Rich Internet Applications - RIAs). Rikkaat verkkosovellukset ovat dynaamisia, nopeita ja interaktiivisia sovelluksia.

AJAX on ryhmä tekniikoita, joiden avulla voidaan rakentaa rikkaita verkkosovelluksia. AJAX on vain yksi malli muiden joukossa, mutta se on saavuttanut muihin verrattuna hypemäisen suosion. AJAX on pop ja kaikki puhuvat siitä, mutta kuinka moni oikeasti tietää, mikä AJAX on. AJAX-hype otti siivet alleen vuonna 2005 ja siitä julkaistiin seuraavana vuonna useita kirjallisia teoksia. Suomenkielisiä teoksia on markkinoilla kuitenkin vielä niukalti. Tämä huomio herätti kipinän kirjoittaa AJAXista suomeksi.

Opinnäytteeni poikkeaa tyypillisestä IT-alan kirjasta. Pyrkimykseni ei ole opettaa lukijoita koodaamaan rikkaita verkkosovelluksia AJAX-tekniikoilla, vaan selvitän teoreettisesti, mistä AJAXissa on kyse ja mihin se oikein soveltuu. Käytän työssäni paljon havainnollistavia esimerkkejä, jotta opinnäytteeni on mahdollisimman selkeä. En suuntaa työtäni ohjelmoijaguruille, vaan lukijoille, jotka ovat kiinnostuneita verkkosovelluksista ja tuntevat WWW-sivujen perustoimintaperiaatteen.

Aloitin työni selvittämällä, mikä AJAX on, mistä tekniikoista se koostuu ja mihin näitä tekniikoita AJAX-sovelluksissa tarvitaan. Vaikeasti hahmotettavan teknisen termistön vuoksi tein työhöni sanaston (liite 1). Sanaston termit on alleviivattu niiden esiintyessä tekstissä ensimmäisen kerran.

AJAX ei ole paistatellut vain loisteessa vaan on saanut osakseen myös paljon kritiikkiä. Monet ovat arvostelleet sen heikkouksia. SWOT-analyysin avulla selvitän AJAXin suosion takana olevat vahvuudet ja kritiikkiä saaneet heikkoudet. Annan lukijalle myös neuvoja heikkouksien välttämiseksi. Esimerkkien avulla tuon esille niin hyviä kuin huonojakin AJAX-sovelluksia. Tarkoitukseni on havainnollistaa, minkälaisiin sovelluksiin AJAX parhaiten soveltuu.

Lopuksi teen AJAXille tulevaisuuden vision. Selvitän mihin suuntaan se on menossa ja minkälaisia suuntamerkkejä siitä on havaittavissa. Onko AJAXissa potentiaalia kiintotähteyteen vai jääkö se pelkäksi tähdenlennoksi?

2 KOHTI RIKKAAMPAA KÄYTETTÄVYYTTÄ

We'll have rich, desktop-like applications that have migrated to the Web, thanks to Ajax, Flex, Flash, Laszlo, and whatever else comes along.

- Dan Saffer, senior interaction designer for Adaptive Path.

Kaikki, jotka vähääkään seuraavat internetin kehittymistä, tietävät termin Web 2.0. Se on tämän hetken kuuma puheenaihe verkkosuunnittelijoiden, yritysten ja alan opiskelijoiden keskuudessa. Se on ilmiö, josta käydään kahvipöytäkeskusteluja ja josta väännetään kättä kansainvälisissä konferensseissa. Web 2.0 ei ole terminä uusi, sillä O'Reilly Media käytti sitä ensimmäisen kerran jo vuonna 2004. Termillä viitataan toisen polven verkkosovelluksiin. Tällaisia uuden sukupolven sovelluksia ovat erilaiset verkostoitumissivustot, wikit, viestintätyökalut ja yhteisöllisesti tuotetut avoimet sivustot. Erityinen paino on yhteistyöllä ja tiedon jakamisella muiden käyttäjien kanssa.

Toinen painopiste Web 2.0:n mukaisissa sovelluksissa on käytettävyyden parantaminen. Tavoitteena on tehdä verkkosovelluksista yhtä käytettäviä kuin työpöydän ohjelmista. Nämä käytettävyydeltään rikkaammat sovellukset tunnetaan alan piireissä nimellä RIA (Rich Internet Application). Ne voidaan toteuttaa useammalla eri mallilla tai tekniikalla, mutta Web 2.0 -tulkinnan mukaan ihanteellisinta olisi tehdä ne avoimiin standardeihin perustuvalla ja Googlen mukana suureen suosioon nousseella AJAXilla.

Dynaamisten ja interaktiivisten sivustojen tekemiseen on käytetty jo pidemmän aikaa erilaisia tuttuja tekniikoita, kuten esimerkiksi JavaScriptiä, XML:ää ja HTML:ää. Nyt näitä samoja tekniikoita käytetään uuden sukupolven verkkosovellusten tuottamiseen. Ensimmäisen kerran nämä tekniikat yhdistettiin yhden nimen alle Applen Developer-sivustolla julkaistussa iframea koskevassa artikkelissa. Tuolloin tekniikoita kutsuttiin nimellä Remote Scripting, mutta nimi ei saavuttanut yleisön keskuudessa suurta suosiota. (Zakas & al. 2006, 13.) Vuonna 2005 Adaptive Pathin johtaja Jesse James Garrett nimesi tekniikkaryhmän uudelleen termillä AJAX. Hollantilaisen jalkapallojoukkueen energisyyteen ja kreikkalaisten sankarien hulluuteen viittaava nimi juontaa juurensa sanoista Asynchronous JavaScript and XML. (Crane & al. 2006, 4.)

AJAX-sovellusten käyttämät tekniikat ovat olleet käytössä jo 90-luvun puolivälistä, mutta niiden yhteiskäyttö AJAXina tuli mahdolliseksi vasta tietokoneiden ja internetin kehittyttyä tarpeeksi. Googlen varapuheenjohtaja Adam Bosworth kertoo eWEEK-lehden haastattelussa, että AJAX olisi syntynyt huomattavasti aikaisemmin, jos mikropiirit olisivat olleet nopeampia ja laajakaista olisi ollut laajemmin käytössä. Tekniikan parantuessa ihmiset alkoivat käyttää verkkosovelluksia jokapäiväisessä elämässään sähköpostin, kalenterin, kaupankäynnin ja verkostojen luomisen muodossa. Oli syntynyt tarve rikkaammille sovelluksille ja tekninen pätevyys entisten tekniikoiden synergialle. (Taft 2007.)

AJAXin kumouksellisin piirre on sen kyky nopeuttaa käyttäjän ja verkkosovelluksen välistä interaktiota. Darryl Taft on artikkelissaan *Google's Bosworth: Why AJAX Failed (Then Succeeded)* sitä mieltä, että hyvän verkkotyökalun pitäisi reagoida käyttäjän toimintaan alle puolessa sekunnissa. Perinteiset WWW-sivut ovat suurten latausmäärien vuoksi hitaita ja epäkäyttäjystävällisiä. AJAX-sovelluksissa latausajat alittavat helpommin tuon puolen sekunnin rajan pienemmän datamäärän vuoksi. (Peltomäki 2006, 296.) Edistyksellisten ominaisuuksiensa ansiosta AJAX-sovellukset kurovat kiinni tietokoneohjelmien ja verkkosovellusten käyttöliittymien välistä kuilua (Crane & al. 2006, 3).

3 AJAXIN TOIMINTA

Ajax isn't a technology. It's really several technologies, each flourishing in its own right, coming together in powerful new ways.

- Jesse James Garrett, president of Adaptive Path

AJAX ei ole ohjelmointikieli. Se yhdistää monen eri teknologian ominaisuuksia käyttäen niitä uudella mielenkiintoisella tavalla. (Crane & al. 2006, 3.) AJAX osaa hyödyntää useita tekniikoita, mutta vain kolme niistä on pakollisia – HTML / XHTML, DOM ja JavaScript (Zakas & al. 2006, 7). Taulukossa 1 esitetään yleisimmät AJAX-tekniikat.

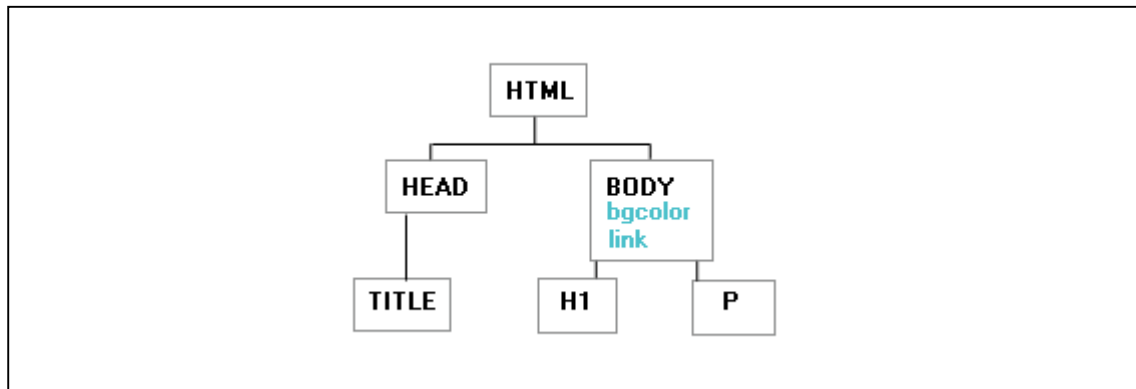
Taulukko 1. Yleisimpien AJAX-tekniikoiden esittely.

AJAX-tekniikat	
TEKNIikka	ROOLI AJAXissa
XHTML	WWW-sivun sisällön ja rakenteen esitystapa
CSS	WWW-sivun tyylimuotoilu
XML	datasisällön tallennusmuoto
DOM	sivun rakennetta kuvaavan solmupuun muodostaminen
XMLHttpRequest	JavaScript-olio, joka hakee tiedon <u>asynkronisesti</u> palvelimelta
JavaScript	<u>skriptikieli</u> , jolla tehdään toimintoja suorittavia ohjelmia

Taulukossa ei ole mainittu kaikkia tekniikoita, joita AJAX-sovelluksissa voidaan käyttää. Listasta puuttuvat esimerkiksi kaikki palvelinpuolen tekniikat, joita ovat mm. PHP ja Ruby on Rails. Taulukon tekniikat ovat lähes kaikki myös korvattavissa muilla tekniikoilla, XHTML HTML:llä, XML JSON:lla ja Google Web Toolkitin myötä myös AJAX-sovellus pystytään kirjoittamaan Javalla, jonka Googlen työkalu kääntää JavaScriptiksi. Tässä opinäytteessä keskitytään vain taulukossa 1 esitettyihin tekniikoihin, sillä ne ovat World Wide Web Consortiumin (W3C) avoimia suosituksia.

3.1 AJAX-sovelluksen ulkoasu

Perinteisesti WWW-sivut koostetaan HTML-merkintäkielellä (HyperText Markup Language), joka on tarkoitettu erityisesti hypertekstidokumenttien toteuttamiseen. Merkitseminen rakentuu puumaisesti ja koostuu erilaisista elementeistä ja niitä kuvailevista attribuuteista. Kuvassa 1 esitetään esimerkki yksinkertaisesta HTML-puumallista, joka koostuu HTML-, head-, body-, title-, h1- ja p-elementeistä sekä body-elementtiä kuvailevista attribuuteista bgcolor ja link. HTML-kielellä muotoillaan dokumentin tekstiä ja ulkoasua, kerrotaan elementtien sijainti dokumentissa ja viitataan muihin WWW-dokumentteihin. Kielen avulla kootaan elementeistä kokonaisuus, jonka internetselain tulkitsee WWW-sivuksi. (Linjama 2001, 62–63.)



KUVA 1. HTML-puumalli

AJAX-sovelluksessa WWW-sivu toteutetaan yleensä XML-kielen kielioppia noudattavalla XHTML-merkintäkielellä, joka muodostuu sanoista eXtensible Hypertext Markup Language. XML-sidoksen vuoksi XHTML noudattaa perinteistä HTML-kieltä huomattavasti tiukempia kielioppisääntöjä, mikä edesauttaa sivujen yhdennäköisyyttä julkaistaessa niitä eri selaimilla. Tarkan kieliopin avulla vältetään myös formatointi- eli dokumentin esitystapaan liittyviltä ongelmilta ja XHTML-dokumentin sisällöstä voidaan antaa rakennekomentojen avulla meta-tietoa. (Linjama 2001, 24, 854.) XML-kielestä ja sen kielioppisäännöistä kerrotaan tarkemmin kohdassa 3.2.

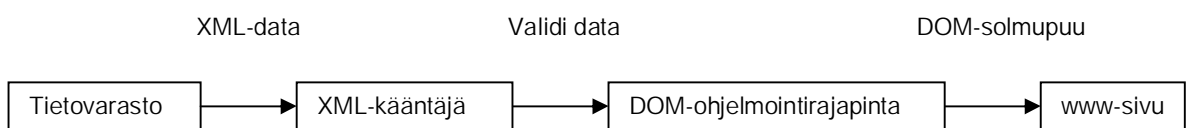
XHTML-merkintäkielellä voidaan määrittää myös sivun ulkoasua, vaikkei sitä siihen ole tarkoitettu. AJAX-sovelluksissa ulkoasu määritetään CSS-tyylitiedostolla (Cascade Style Sheets). Tyylitiedoston määritysten avulla voidaan muuttaa WWW-sivuston ulkoasua nopeasti ja helposti. Tehokkaimmin se tapahtuu linkittämällä XHTML-dokumenttiin ulkoinen tyylitiedosto. (Linjama 2001, 501–501.) Tyylien avulla XHTML-elementtien sijoittaminen haluttuun kohtaan sujuu kätevämmän ja tarkemmin kuin XHTML-kiellä. Dynaamisuutta sivustoon saadaan muokkaamalla elementtien sijoittelua JavaScript-käskyillä. W3C:n laatimat CSS-standardit (CSS 1 ja CSS 2/2.1) lisäävät tyylien käyttömahdollisuuksia ja edesauttavat selainten tukea tyyliille. Standardeista huolimatta CSS:n määrittelyssä ja toteutuksessa on vielä paljon puutteita ja selainkohtaisia virheitä. (Peltomäki & Nykänen 2006, 260.)

3.2 Tiedon kuvaaminen

W3C suosittaa kirjoittamaan AJAX-sovelluksissa käytetyn datan XML-muotoon, mutta data voidaan palauttaa selaimelle myös muilla merkintäkielillä tai pelkkänä tekstinä. Extensible Markup Language (XML) on menetelmä rakenteisen tekstitiedoston toteuttamiseksi ja sitä käytetään aina yhdessä muiden määritysten ja erilaisten valmisohjelmistojen kanssa. Se on merkintäkieli, joka sisältää ”tietoa tiedosta”, eikä sillä pyritä kuvaamaan dokumentin ulkoasua, kuten luvussa 3.1 esitellyillä XHTML-kielellä ja CSS-tiedostoilla tehdään. (Nykänen 2001, 2.)

XML-merkintäkieli on luonteeltaan metakieli ja sen avulla voidaan määritellä, laajentaa ja yhdistellä sovelluskohtaisten tietojen kuvaamiseen tarkoitettuja muita merkintäkieliä. Esimerkiksi HTML-kielestä on määritelty XML:n avulla luvussa 3.1 esitelty XHTML-merkintäkieli. Verkkosovelluksissa XML-tiedosto toimii relaatiotietokannan kaltaisena yleiskäyttöisenä tietovarastona. (Nykänen 2001, 2.)

XML on ensisijaisesti AJAX-sovellusten tekninen perusta, eikä XML-tekstiä ole sellaisenaan tarkoitettu käyttäjien luettavaksi. XML-tiedosto toimii tietovarastona, josta noudetaan dataa erilaisten JavaScript-käskyjen avulla. (Nykänen 2003.) Kuva 2 esittää XML-datan matkan tietovarastosta käyttäjän nähtäväksi.



KUVA 2. XML-datan matka WWW-sivulle.

Kuvan 2 tietovarastosta lähetetään XML-kääntäjälle dataa. Kääntäjä tarkistaa datan kieliopin ja lähettää validin datan DOM-ohjelmointirajapinnalle, jossa data jäsennetään DOM-solmupuuksi. WWW-sivulla oleva XML-sovellus (esim. XHTML) tulkitsee solmupuun JavaScriptin avulla ja julkaisee tiedon WWW-sivulla. DOM-ohjelmointirajapinnan toiminta kuvataan tarkemmin alaluvussa 3.3.

Käytännössä merkitseminen tarkoittaa tekstin kuvaamista informaatiolla. Tämän informaation avulla dokumenteista saadaan järjestelmällisempiä ja niiden sisältöä on helpompi muokata. Merkintä perustuu hierarkkiseen elementtirakenteeseen, jonka elementtejä kuvataan attribuuteilla. XML-kieli koostuu sääntöjen määrittelyyn tarkoitettu kieliopista ja sovelluskohtaisesta sanastosta. Kielioppi itsessään jakautuu kahteen osaan: merkintään ja merkkidataan. Sanasto puolestaan koostuu elementtien ja attribuuttien nimistä. Sanastoa ohjaa sanaston oma kielioppi, joka määrittää dokumentin tyyppin. (Nykänen 2003.)

XML on merkintäkielen lisäksi myös tyyppimäärittelykieli, jonka avulla voidaan erottaa toisistaan esimerkiksi kirjeet ja bussiaikataulut mallintamalla niiden loogista rakennetta (Nykänen 2001, 2). Dokumentin tyyppi määrittelee XML-dokumentin elementtirakenteen ja se voidaan mieltää tiedonsiirron rajapintana kahden kommunikoivan tahon välillä. Tyyppin määrittäviä sanaston kielioppeja ovat esimerkiksi Document Type Definitionin (DTD) ja W3C:n suosittelema XML Schema. (Peltomäki & Nykänen 2006, 29.)

Laajempaan kokonaisuuteen XML:stä käytetään nimitystä XML-perhe. Tällä viitataan joukkoon teknisesti yhteensopivia määrittäviä ja työkaluja, joiden avulla voidaan merkkauksen lisäksi määrittellä dokumentin tyyppi, paloitella dokumentti, erotella dokumentin rakenne ja esitystapa toisistaan, kuvailla ja yhdistää tietoa ja käsitellä XML-dokumentteja yhtenäisesti DOM-rajapinnalla. (Nykänen 2003.) XML-perheen ytimen eli standardiperheen muodostavat XML, XML Schema, XML Namespaces ja DOM (Peltomäki & Nykänen 2006, 13).

Kuten jo alaluvussa 3.1 kerrottiin, XHTML käyttää XML:n kielioppia. XML-dokumentin kieliopin täytyy olla täysin oikein tai selain antaa virheilmoituksen. Perinteisissä HTML-dokumenteissa selaimet sallivat suuriakin kielioppivirheitä. (Linjama 2001, 806.) XML-dokumentin kielioppisäännöt on listattu taulukkoon 2.

Taulukko 2. XML-dokumentin kielioppisäännöt (Heinonen 2007).

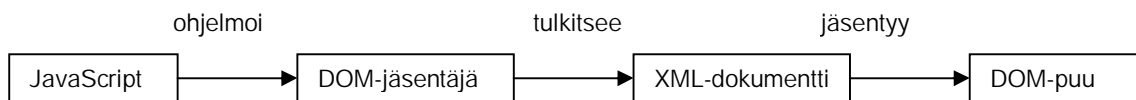
1.	Dokumentilla on aina juurielementti ja niitä on vain yksi.
2.	Kaikilla elementeillä on sekä aloitus- että lopetustagi.
3.	Dokumentin rakenne noudattaa puumallia.
4.	Nollatoleranssi merkintäkieliopillisia virheitä kohtaan.

3.3 Datan tulkitseminen solmupuuksi

Document Object Model (DOM) on W3C:n suosittama ohjelmointirajapinta (API).

DOM on alusta- ja ohjelmointikieliriippumaton ja sitä voidaan käyttää millä ohjelmointikielellä tahansa. DOMin käyttöönotto on helppoa, sillä kaikilla selaimilla on omat DOM-toteutuksensa, joiden avulla WWW-sivut ovat ohjelmointikielien saavutettavissa. (The Mozilla Organization 2003.)

Palvelimen lähettämästä XML-dokumentista saadaan tietokoneen muistiin rakennettu DOM-puu jäsentämällä dokumentti ohjelmointikielellä luodun DOM-jäsentäjän avulla (Lipitsäinen 2002). Kuvassa 2 kuvattiin XML-datan muuttuminen DOM-puuksi DOM-rajapinnalla. Kuva 3 havainnollistaa tarkemmin, mitä DOM-rajapinnalla tapahtuu.

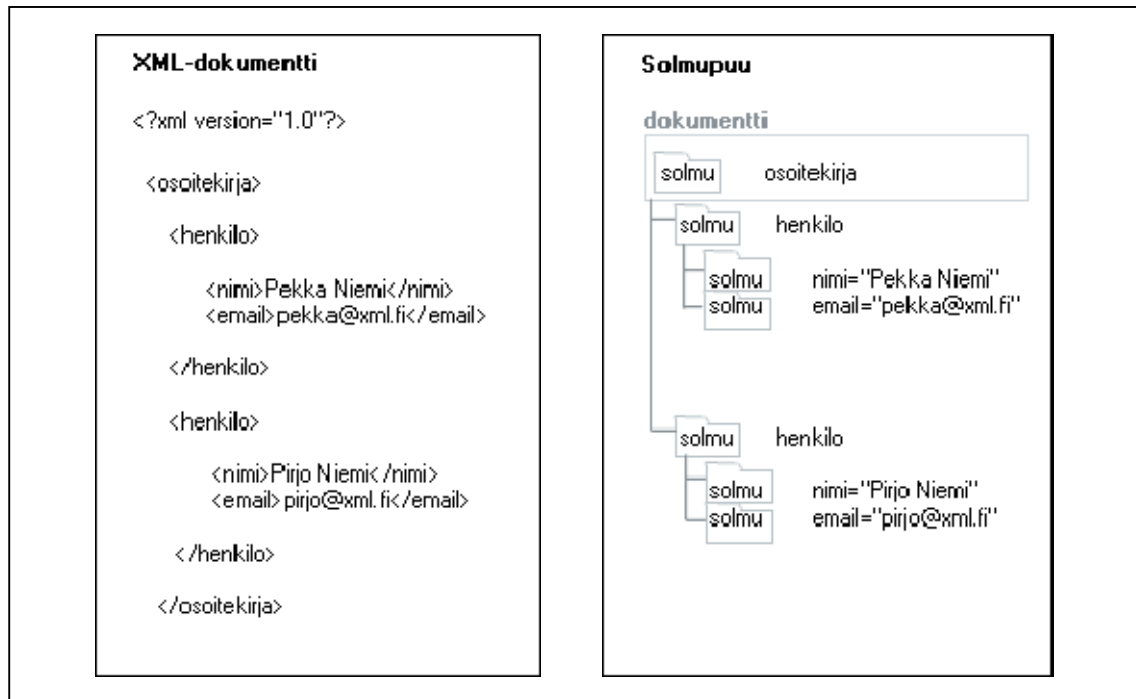


KUVA 3. DOM-puun jäsentäminen XML-dokumentista.

JavaScriptin avulla ohjelmoidaan DOM-jäsentäjä, joka tulkitsee XML-dokumentin eli XML-käsittelijän lähettämän validin datan. Tämän jälkeen jäsentäjä jäsentää datan DOM-puuksi, jonka XHTML tulkitsee ja julkaisee WWW-sivulla.

DOM esittää dokumentin hierarkkisen solmupuuna, joka koostuu vanhempi-, lapsi-, sisar-, esivanhempi- ja jälkeläissolmuista. Solmut mahdollistavat viittauksen elementteihin näiden keskinäisten suhteiden avulla. Hierarkian ylimpinä ovat aina selainta kuvaava window-olio ja heti sen alapuolella oleva dokumentti-olio, joka on selaimelle ladattava HTML-sivu. Sukulaisuus-metodilla voidaan ilmaista jokaiselle

dokumentille uniikki rakenne, jossa dokumentin kaikki elementit tunnistetaan ja ilmaistaan osana rakennetta. (Campbell & Darnell 1997, 50–51.) Kuvassa 4 on esitetty esimerkki XML-dokumentin DOM-puumallista.



KUVA 4. DOM-solmupuu

Kuvan 4 vasemman puoleisessa laatikossa on esitetty XML-osoitekirjan koodi ja oikealla siitä on tehty DOM-solmupuu. Solmupuu koostuu dokumentti-solmusta ja sen lapsisolmusta (osoitekirja). Dokumentti-solmu toimii isovanhempisolmuna henkilösolmulle, jolla on vuorostaan kaksi lapsisolmua (nimi ja email).

Jokainen WWW-sivun elementti on osa dokumentin DOM-puuta ja niitä voidaan muokata solmupuun ja JavaScriptin avulla. Rajapintaa hyödyntämällä HTML- ja XML-dokumentteihin upotettu JavaScript pystyy luomaan uusia dokumentteja, navigoimaan dokumentin rakenteessa, lisäämään, poistamaan ja muuttamaan elementtejä, ottamaan käsittelyyn dokumentin osia ja paljon muuta. (The Mozilla Organization 2003.)

Dokumenttimalli määrittelee elementtien välisen kommunikaation ja sen miten JavaScript-ohjelman avulla voidaan viitata näihin elementteihin. AJAX-sovellukselle DOM on elintärkeä, sillä ilman sitä JavaScriptillä ei pystyttäisi vaikuttamaan WWW-sivun elementteihin. (Peltomäki & Nykänen 2006, 98–99.)

DOM mahdollistaa JavaScriptillä ohjelmoitujen muutoksien teon myös dokumentin tyylimääriytyksiin. DOMin tyylimääriytystuen avulla elementtien ulkoasua tai niiden sijaintia WWW-dokumentissa voidaan muuttaa. Nämä ominaisuudet antavat käyttäjälle muun muassa mahdollisuuden personoida WWW-sivua raahaamalla sivun elementtejä haluamaansa paikkaan tai muuttamalla niiden ulkoasua. (Peltomäki & Nykänen 2006, 260.)

3.4 Toimintojen toteuttaminen

AJAX-sovelluksissa käytetään JavaScript-kieltä sen helppouden ja nopeuden vuoksi. JavaScriptin avulla on mahdollista tehdä ohjelmia, joiden ansiosta käyttäjä kokee välittömän vuorovaikutuksen sivuston kanssa. (Peltomäki & Nykänen 2006, 97.)

JavaScript on selainpohjainen skriptikieli, jonka avulla tuotetaan dynaamisia verkkosivuja. JavaScript on suunnattu niin ammattiohjelmoijille kuin harrastelijoillekin, minkä vuoksi kielen vaatimukset eivät ole tiukkuudeltaan samat kuin sovelluskehityskielissä (esim. C++ ja Java). Sovelluskielistä JavaScript eroaa myös tulkattavuutensa ansiosta, sillä esimerkiksi Java ja C++ -kielet on käännettävä ennen niiden suorittamista. Tulkattavuus helpottaa ja nopeuttaa ohjelmistokehitystä. Ohjelma suoritetaan aina selaimessa tai muussa JavaScript-tulkin sisältävässä sovelluksessa. (Peltomäki & Nykänen 2006, 90, 93–94, 96–97.)

JavaScript-syntaksi pohjautuu osittain Javaan, mutta tästä huolimatta kielillä on hyvin paljon eroja. JavaScript on Javaa yksinkertaisempi kieli ja se soveltuu huomattavasti pienempien ohjelmien tuottamiseen kuin Java. JavaScript eroaa Javasta myös siksi, ettei sitä varten tarvitse ladata omaa kehitysympäristöä. Java-sovelluksella voidaan vaikuttaa rajalliseen alueeseen HTML-dokumentin sisällä, mutta JavaScriptillä pystytään muokkaamaan koko HTML-dokumenttia. (Peltomäki & Nykänen 2006, 97.)

JavaScript on olio-ohjelmointikieli. Olio-ohjelmoinnin avulla ohjelmat voidaan jakaa itsenäisiksi ja loogisiksi kokonaisuuksiksi, mikä helpottaa ohjelmien ylläpitoa. Olioiksi lasketaan merkkijonojen ja numeroiden lisäksi myös funktiot. JavaScript-kieleen on sisäänrakennettu valmiita olioita, mutta olioita voi tehdä myös itse. (Peltomäki & Nykänen 2006, 107.)

Graafisella käyttöliittymällä havainnollistetaan usein käyttäjän tai järjestelmän tuottamia tapahtumia, joihin JavaScript reagoi. Tapahtumiksi luetaan esimerkiksi hiiren painikkeen painaminen elementin päällä ja kohdistimen siirtäminen syöttökenttään. Tapahtumiin reagoidaan tapahtumankäsittelijöillä, joita ovat esimerkiksi onmouseover, onclick ja onmouseout. Tapahtumamalleilla puolestaan määritellään, minkälaisia tapahtumia toteutetaan, miten tapahtumat välitetään käsittelijöille ja mitä tietoa käsittelijällä on funktion suorittamista varten. (Peltomäki & Nykänen 2006, 110–111.) Taulukko 3 kuvaa JavaScriptin suorittaman tapahtumaketjun, joka alkaa hiiren napin painalluksesta.

Taulukko 3. JavaScriptin toiminnan kulku.

1. Käyttäjä painaa hiiren painiketta
2. Tapahtumamalli määrittää miten tapahtuma välitetään käsittelijälle.
3. Tapahtumakäsittelijä suorittaa funktion.

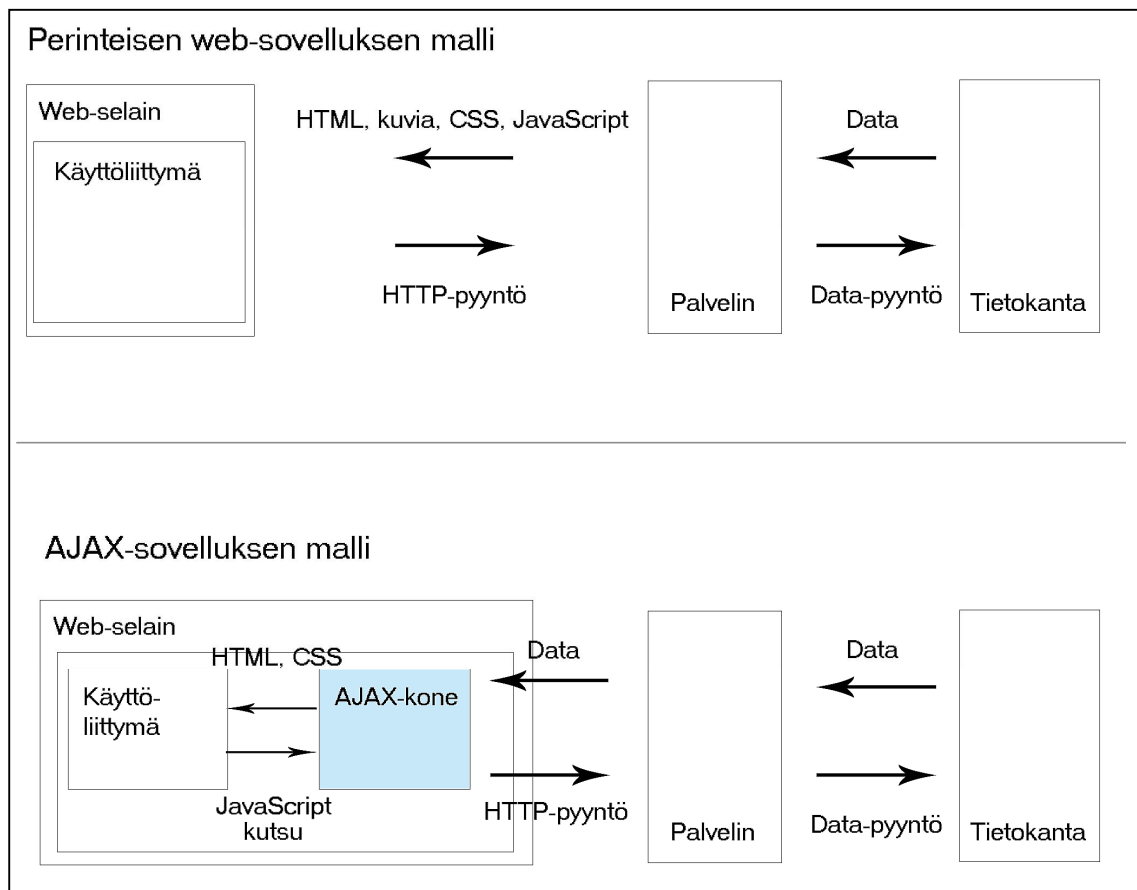
AJAX perustuu paljolti JavaScriptin toimintaan ja sen kykyyn sitoa tarvittavat tekniikat yhteen. XHTML-dokumentin muokkaaminen tapahtuu JavaScriptin ja DOMin yhteistyönä. Palvelimelta palautuvan datan jäsentämiseen tarvitaan JavaScriptiä. Ja ennen kaikkea JavaScript-kieltä tarvitaan AJAXin ytimen eli ”AJAX-koneen” ohjelmoimiseen.

AJAX-tekniikoiden myötä vuorovaikutus sovelluksen ja käyttäjän välillä on parantunut. Erona perinteiseen hitaampaan sovellukseen on se, että lähetetyn informaation määrä on huomattavasti entistä pienempi. Tavallisesti palvelin palauttaa HTML-kieltä, kuvia, tyylejä ja JavaScriptiä, mutta AJAX-sovelluksissa palvelin palauttaa vain dataa. (Zakas & al. 2006, 5.)

Pienempi datamäärä johtuu perinteisen ja AJAX-sovelluksen rakenteellisista eroista. Perinteisessä verkkosovelluksessa selain lähettää pyyntöjä palvelimelle. AJAX-sovelluksiin ohjelmoidaan JavaScriptillä ”AJAX-kone”, joka toimii selaimen ja palvelimen välillä hoitaen pyyntöjen lähettämisen. Tämä XMLHttpRequest-olio on JavaScript-olio eli funktio, jota kutsutaan, kun palvelimelle tarvitsee lähettää pyyntö. XMLHttpRequest-olio tekee pyynnön palvelimelle asynkronisesti, mikä tarkoittaa, ettei selain pysähdy odottamaan vastausta, vaan jatkaa toimintaansa normaalisti. AJAX-

koneen ansiosta AJAX-sovellukset ovat huomattavasti nopeampia ja interaktiivisempia kuin perinteiset sovellukset. (Zakas & al. 2006, 5.)

Käytännössä AJAX toimii seuraavasti: XHTML-sivustoon upotetaan JavaScript-koodia, joka pystyy tarvittaessa hakemaan tietoa palvelimelta XMLHttpRequest-olion avulla. Tämä tiedonhaku tapahtuu asynkronisesti ja piilossa käyttäjältä. Tietoa odotellessa sivusto toimii normaalisti ja käyttäjä voi jatkaa sivun selaamista. Tieto palautuu palvelimelta yleensä W3C:n suosituksen mukaisesti XML-muodossa, mutta palvelimelta voidaan palauttaa myös pelkkää tekstiä. JavaScript lukee tiedon ja päivittää DOMin avulla vain sen osan sivusta, jota muutos koskee. Muu käyttöliittymä pysyy koko ajan muuttumattomana ja käyttäjän ulottuvissa. (Holzner 2006, 11.) Seuraavassa kuvassa havainnollistetaan perinteisen verkkosovelluksen ja AJAX-sovelluksen eroja.



KUVA 5. Perinteisen ja AJAX-sovelluksen toiminnan erot (Zakas & al. 2006, 5).

Kuvassa 5 perinteisen sovelluksen selain lähettää palvelimelle pyynnön, jonka palvelin toteuttaa ja palauttaa suuren määrän tietoa. Pyyntö ei ole asynkroninen, minkä vuoksi selain odottaa sivun latautumista. AJAX-sovelluksessa puolestaan JavaScriptillä on ohjelmoitu AJAX-kone eli XMLHttpRequest-olio, joka lähettää palvelimelle

asynkronisen pyynnön. Palvelin palauttaa vain dataa, minkä vuoksi interaktio on nopeampi kuin perinteisessä sovelluksessa. Asynkronisen pyynnön vuoksi selain ei pysähdy odottamaan latautumista, vaan jatkaa toimintaansa normaalisti.

4 RISUJA JA RUUSUJA

While I can name quite a few AJAX-based Web applications that have fantastic and intuitive interfaces, I can also name several that have gone way over the top with pop-up windows, scrolling menu bars, flashing icons and other elements.

– Jim Rapoza, director of eWEEK Labs.

AJAX ei ole saanut osakseen pelkkää suitsutusta. Sitä on moitittu niin turvallisuuden puutteesta kuin liiallisesta koodimäärästäkin. AJAX jakaa mielipiteitä ja sen vahvuudet saavat kehuja ja heikkoudet huutia.

SWOT-analyysi on yksinkertainen tapa tarkastella AJAXin vahvuuksia ja heikkouksia. Analyysin tavoitteena on saada selkeä kuva AJAXin nykytilanteesta ja tulevaisuuden mahdollisuuksista ja uhista. SWOT-analyysissa samat asiat ovat usein vahvuuksia, heikkouksia, mahdollisuuksia ja uhkia. Tämä johtuu asioiden arvioinnin subjektiivisuudesta. Samassa asiassa nähdään monta eri puolta. (Lindroos & Lohivesi 2004, 35, 217.)

Analyysin alustana käytetään nelikenttää, jossa on eroteltuina positiiviset ja negatiiviset piirteet ja nykyhetki ja tuleva. Nykyhetkeä edustavat ylemmät lohkot (vahvuudet ja heikkoudet) ja tulevaisuutta edustaa nelikentän alaosa (mahdollisuudet ja uhat). (Lindroos & Lohivesi 2004, 218.)

SWOT-analyysin tulosten avulla voidaan päätellä, miten vahvuuksia voidaan käyttää hyväksi, miten heikkoudet muutetaan vahvuuksiksi, miten tulevat mahdollisuudet hyödynnetään ja miten mahdolliset uhat vältetään (Lindroos & Lohivesi 2004, 218).

4.1 AJAXin vahvuudet

AJAXin suosio on seurausta sen perinteistä verkkosovellusta parantavista ominaisuuksista. AJAXin vahvuudet on koottu alla olevaan SWOT-nelikenttään.

SWOT-ANALYYSI

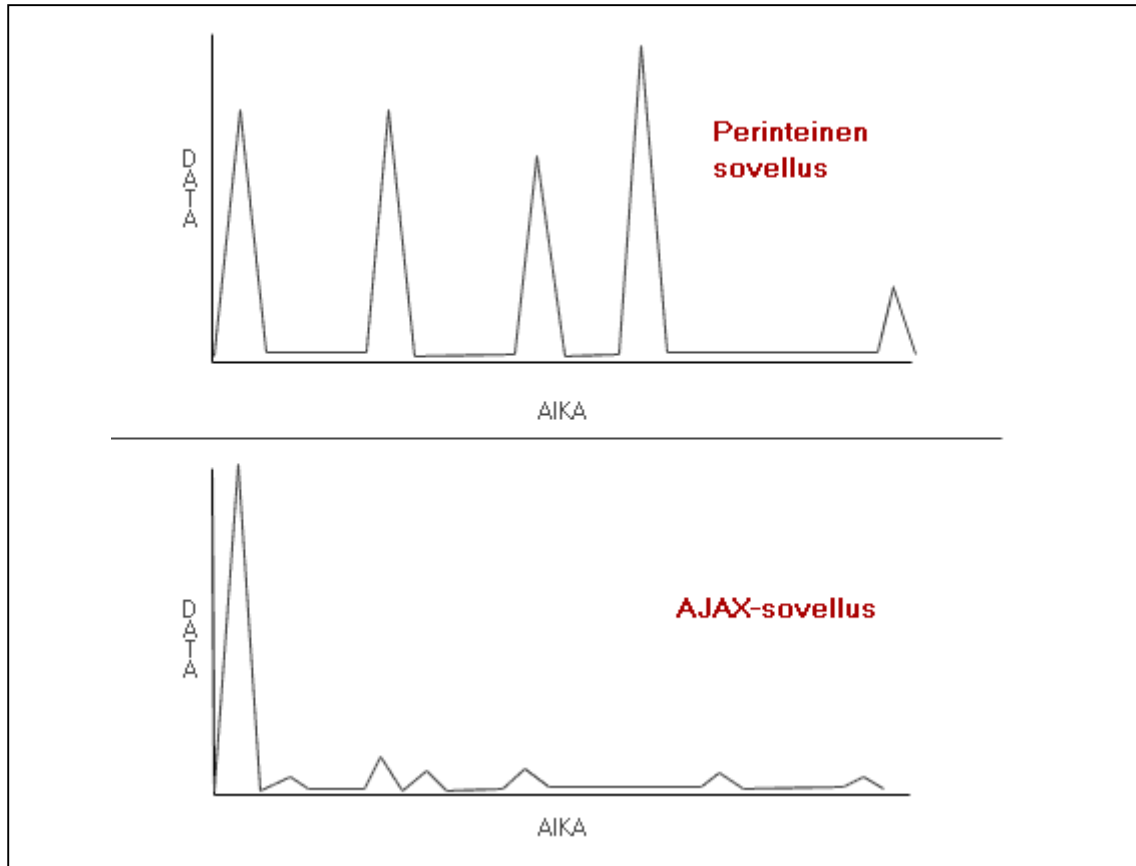
<p>VAHVUUDET</p> <p>Nopeus Avoin standardi Läpinäkyvyys Tietojen tallennus Samankaltaisuus työpöydän ohjelmien kanssa</p>	<p>HEIKKOUEDET</p> <p>Tietoturva Navigoinnin muuttuminen Yhteensopivuus Muistin varaaminen Trendikkyys</p>
<p>MAHDOLLISUUDET</p> <p>Leviäminen Kehittyminen</p>	<p>UHAT</p> <p>Kilpailijat Trendit</p>

KUVA 6. Vahvuudet.

4.1.1 Nopeus

AJAX-keskustelut kääntyvät usein sen erinomaiseen kykyyn nopeuttaa vuorovaikutusta. Nopeus perustuu pyynnöissä lähetettävään pieneen datamäärään ja pyyntöjen asynkronisuuteen.

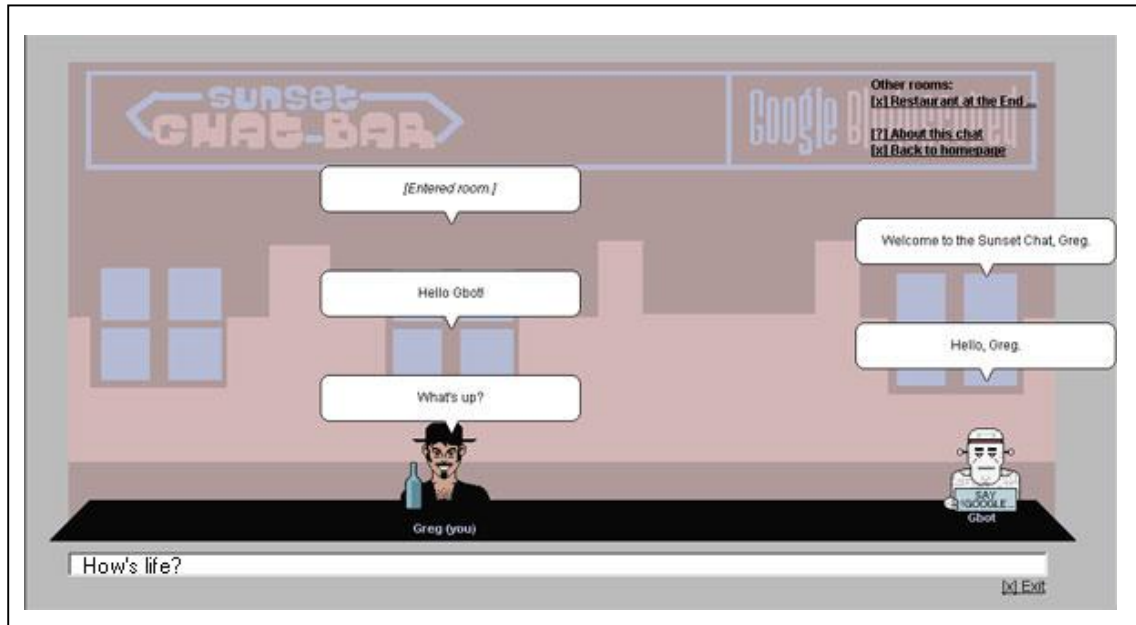
AJAX-sovelluksissa latautumisajan huippu koetaan heti sisäänkirjautumisen yhteydessä (ks. kuva 7). Tässä vaiheessa AJAX-sovellus on yhtä hidaskin kuin perinteinen sovellus, koska molemmissa sovelluksissa ladataan ensin pohjateksti, navigaatio, layout ja sivun sisältö. Sisäänkirjautumisen jälkeen AJAX-sovellusten latausajat pienentyvät radikaalisti. Sivulle ei enää tarvitse ladata niin paljon tietoa ja sisältöä. Perinteinen sivusto jatkaa koko sivun lataamista, mutta uuden sukupolven sovelluksissa ladataan kokonaisen sivun sijasta vain tietty solmu DOM-puusta. (Crane & al. 2006, 19–21.)



KUVA 7. Perinteisen ja AJAX-sovelluksen datamäärät.

Asynkronisten pyyntöjen ansiosta selain ei pysähdy odottamaan palvelimen palauttamaa tulosta pyynnölle. Käyttäjän ei siis tarvitse odotella turhan pitkään sivun päivitystä, vaan hän voi selailla odotellessaan sivua normaalisti. Sivun käyttöliittymä on koko ajan saatavilla. (Crane & al. 2006, 13–14.)

Nopeutensa ja asynkronisuutensa ansiosta AJAX soveltuu hyvin esimerkiksi erilaisten keskustelupalstojen tekniseksi perustaksi. Keskustelu sujuu nopeasti ja katkeamattomana virtana pienen datamäärän vuoksi. Oman lähetetyn viestin latautumisen ajan käyttäjä voi lukea muiden keskustelijoiden viestejä. Yksi esimerkki internetissä olevasta AJAX-tekniikoihin perustuvasta chat-palvelusta on Phillip Lensenin rakentama Sunset Chat.



KUVA 8. Sunset Chat -keskustelufoorumi (Sunset Chat).

Sunset Chatin toiminta on hyvin yksinkertainen ja helposti omaksuttavissa. Käyttäjä valitsee ensin itselleen avatarin ja nimimerkin. Tämän jälkeen hänen avatarinsa ilmestyy Chat Barin tiskille ja keskustelu voidaan aloittaa. Viesti kirjoitetaan baarin alapuolella olevaan tekstiruutuun ja se julkaistaan painamalla enteriä. Viesti ilmestyy ruudulle puhekuplana. Avatarien ansiosta Sunset Chatissä on hauska sarjakuvamainen tunnelma. palvelun interaktio on nopeaa ja toimivaa. Ainoana haittapuolena on se, ettei tiskille mahdu kuin kymmenen keskustelijaa kerrallaan.

4.1.2 Avoin standardi

AJAX-sovelluksissa käytetyt tekniikat ovat avoimia suosituksia eli kuka tahansa pääsee muokkaamaan ja kehittämään niitä. Koodin avoimuus ja tekniikoiden ilmainen käyttö lisäävät AJAX-sovellusten kehittymistä. Kaikki halukkaat voivat esteettä parantaa ohjelmien koodia.

AJAX-sovelluksien tekniikat ovat W3C:n suosituksia. Standardimääritelmä edesauttaa suosittujen ohjelmistovalmistajien ohjelmistojen, ohjelmointikielten ja kehitysvälineiden AJAX-tuen muodostumista. Tämän ansiosta yhteensopivuusongelmat pienenevät ja sovellukset toimivat paremmin erilaisissa ympäristöissä ja selaimissa. (Peltomäki & Nykänen 2006, 297–298.)

4.1.3 Läpinäkyvyys

AJAXin läpinäkyvyyttä voidaan pitää niin positiivisena kuin negatiivisenakin piirteenä. Se viittaa AJAX-koneen kykyyn tehdä palvelimelle pyyntöjä käyttäjän huomaamatta. Positiivista läpinäkyvyys on siksi, ettei jatkuvilla pyynnöillä häiritä käyttäjää. Pyynnot suoritetaan piilossa käyttäjältä, mutta niiden tulokset tulostuvat ruudulle. Hyviä esimerkkejä ovat hakukoneet, jotka ehdottavat käyttäjän näppäilemille kirjaimille jatkoa ja verkkolomakkeet, joissa tiedot tarkistetaan reaaliaikaisesti. Lomakkeiden syötteitä validoidaan sitä mukaan, kun käyttäjä niitä tekee. Syötteiden avulla voidaan myös muuttaa lomakkeen sisältöä dynaamisesti. (Peltomäki & Nykänen 2006, 298.)

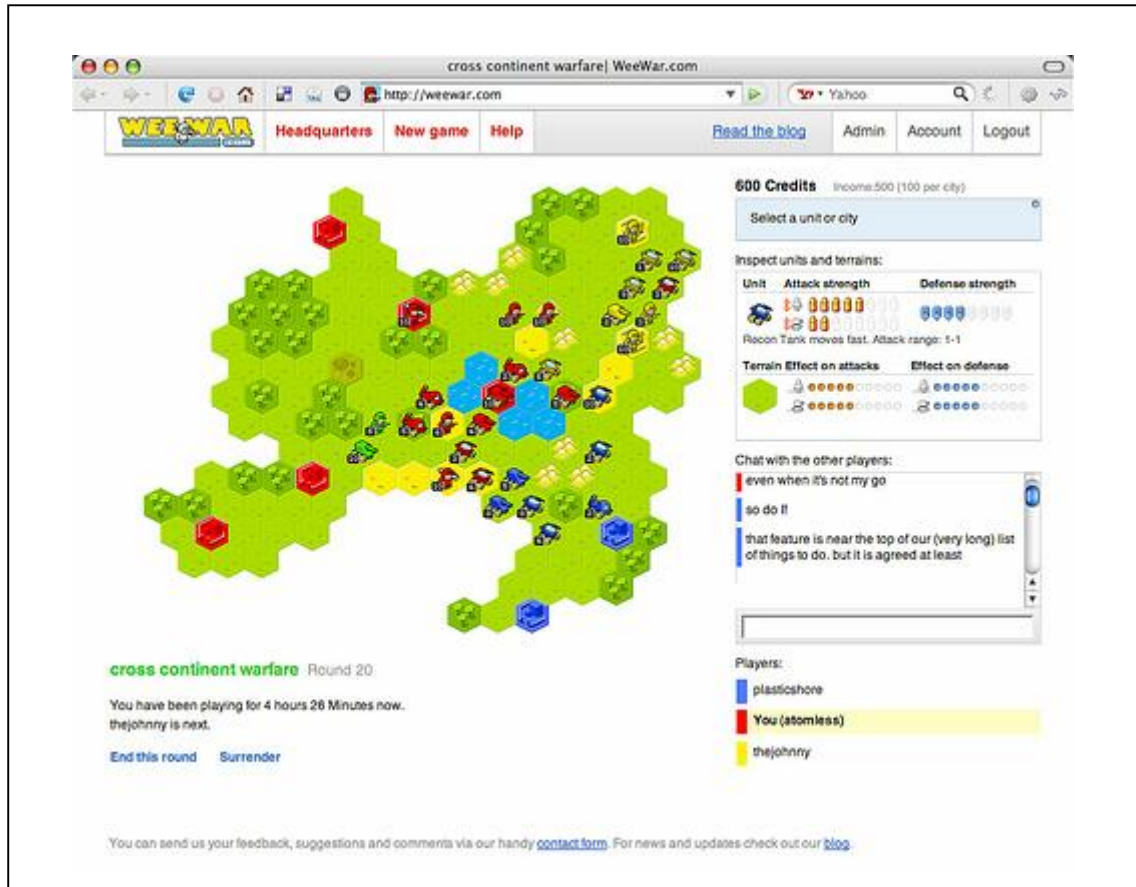
Uusi matkahaku-palvelu Kayak käyttää hyväkseen läpinäkyvyyttä. Ohjelma ehdottaa käyttäjän näppäilemän kaupungin tai lentokentän nimelle jatkoa, minkä ansiosta käyttäjän ei tarvitse kirjoittaa haluamaansa kohdetta loppuun tai huolehtia sanan oikeasta kirjoitusmuodosta. Tämä toiminta on havainnollistettu kuvassa 9. Monissa perinteisellä tavalla toteutetuissa matkahauissa hae-komennon jälkeen siirrytään uudelle sivulle, jossa varmistetaan, että käyttäjä on hakemassa juuri sitä lentokenttää minkä kirjoitti. Kayak toimii asynkronisuutensa vuoksi huomattavasti nopeammin.

KUVA 9. Kayakin lentohakupalvelu (Kayak).

4.1.4 Tietojen tallennus

Perinteisillä WWW-sivustoilla liikutaan sivulta toiselle. Siirryttäessä uudelle sivulle selain poistaa vanhan sivun. Tämän vuoksi istunto tuhoutuu, kun sivustolta kirjaudutaan ulos tai selain suljetaan. Kirjauduttaessa seuraavalla kerralla ohjelmaan sisälle ei päästä enää kiinni edellisen istunnon tietoihin. AJAX-sovelluksessa selaimelle toimitetaan sisäänkirjautumisen jälkeen totuttua monimutkaisempi JavaScript-dokumentti, joka seuraa mukana koko istunnon ajan. Dokumentti muuttuu käyttäjän vuorovaikutuksen myötä, sillä se osaa vastata käyttäjän suorittamiin toimintoihin. JavaScript-dokumentti voi tallentaa tietoa koko istunnon ajalta, minkä vuoksi käyttäjä pääsee käsiksi tallentuneisiin tietoihin myös seuraavilla istunnoilla. (Crane & al. 2006, 18.)

JavaScript-dokumentin kykyä tallentaa tietoa käytetään hyväksi mm. erilaisissa verkkopeleissä. Tiedon tallentuminen tekee peleistä joustavampia. Pelin voi tarvittaessa jättää kesken ja palata myöhemmin jatkamaan sitä samasta kohtaa kuin mihin pelin aiemmin keskeytti. Hyvä esimerkki verkkopelistä on strategiapeli Weewar.com.



KUVA 10. Weewar.comin pelinäköymä (Weewar.com).

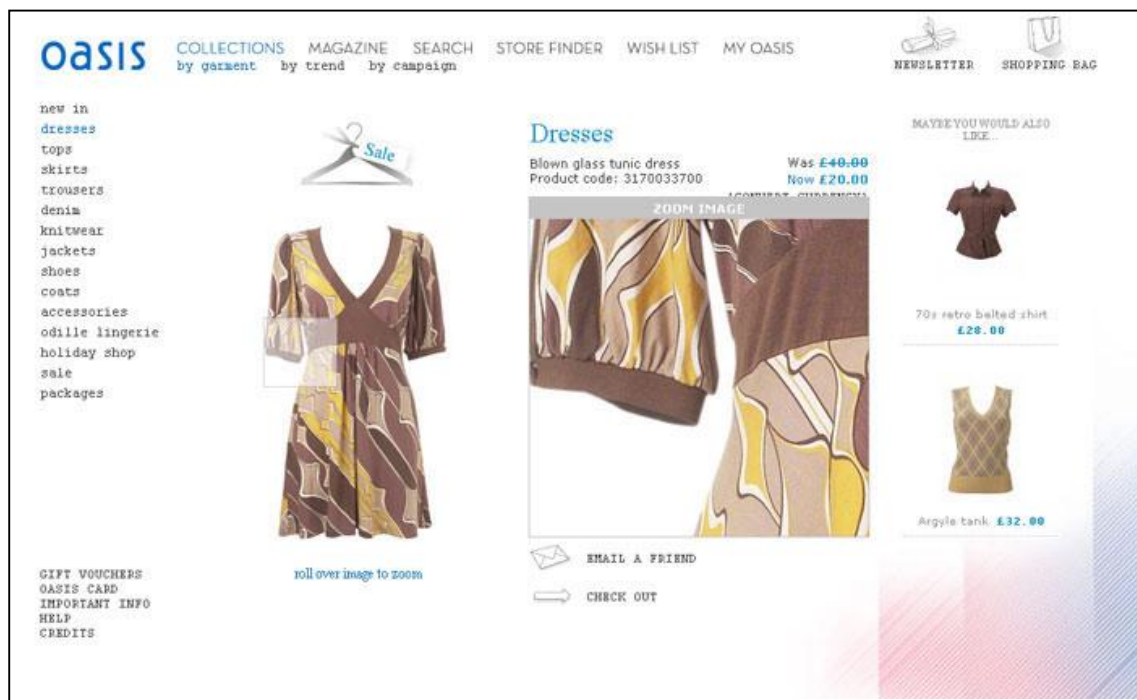
WeeWar.comissa voi pelata kerrallaan kuusi pelaajaa. Käytettävyydeltään pelin tekee erinomaiseksi se, ettei selaimessa toimiva peli vaadi minkäänlaista lataamista ja se pystyy tallentamaan tietoa pelin aikana. Selainikkunan voi huoletta sulkea menettämättä peliä. Tietojen tallentumisen ansiosta peliä voi jatkaa samasta kohdasta, johon sen on viimeksi jättänyt. Pelihahmoja liikutetaan näppäimistön avulla. Teknisesti siirrot tapahtuvat käskyttämällä CSS-tiedostoa JavaScriptin ja DOM-puun avulla.

4.1.5 Samankaltaisuus työpöydän ohjelmien kanssa

AJAX on hyvä keino tuottaa rikkaita verkkosovelluksia, koska sen avulla voidaan siirtää useita työpöydän ohjelmille tyypillisiä ominaisuuksia verkkoon. Näiden ominaisuuksien avulla voidaan käytettävyyttä parantaa mm. informaation esittämisen, automaattisen päivityksen, personoinnin ja havainnollistamisen muodossa.

Kaupallisissa palveluissa suosittu uusi toiminto on drag and drop -menetelmällä tehty ostoskori. Käyttäjä voi raahata haluamansa tuotteen ostoskoriin, jonka sisältö päivittyy sitä mukaan kun tuotteita sinne raahataan. Perinteisiin sovelluksiin verrattuna käyttäjä kokee ostosten lisäämisen huomattavasti konkreettisempänä ja nopeampana toimintana. AJAXia voidaan käyttää myös kaupallisen tuotteen havainnollistamiseen. Perinteisesti tuotekuvat ovat staattisia ja tilanpuutteen vuoksi pieniä. AJAX-tekniikoiden avulla voidaan tuotetta zoomata esimerkiksi erillisen kelluvan ikkunan avulla.

Brittiläisen Oasis-vaateketjun sivuilla voidaan vaatetta suurentaa liikuttamalla hiirtä sen päällä. Toimintoa havainnollistetaan kuvassa 11. Kuvan vaatetta voi tarkkailla suurennetuissa näkymässä, joka avautuu kelluvaan ikkunaan. Kun hiiri viedään pois kuvan päältä, poistuu myös kelluva ikkuna näkyvistä. Perinteisessä sovelluksessa vaatteesta saa suurennetun kuvan usein linkin kautta. Linkistä latautuu kuitenkin uusi sivu, mikä on AJAXin ratkaisua hitaampi vaihtoehto.

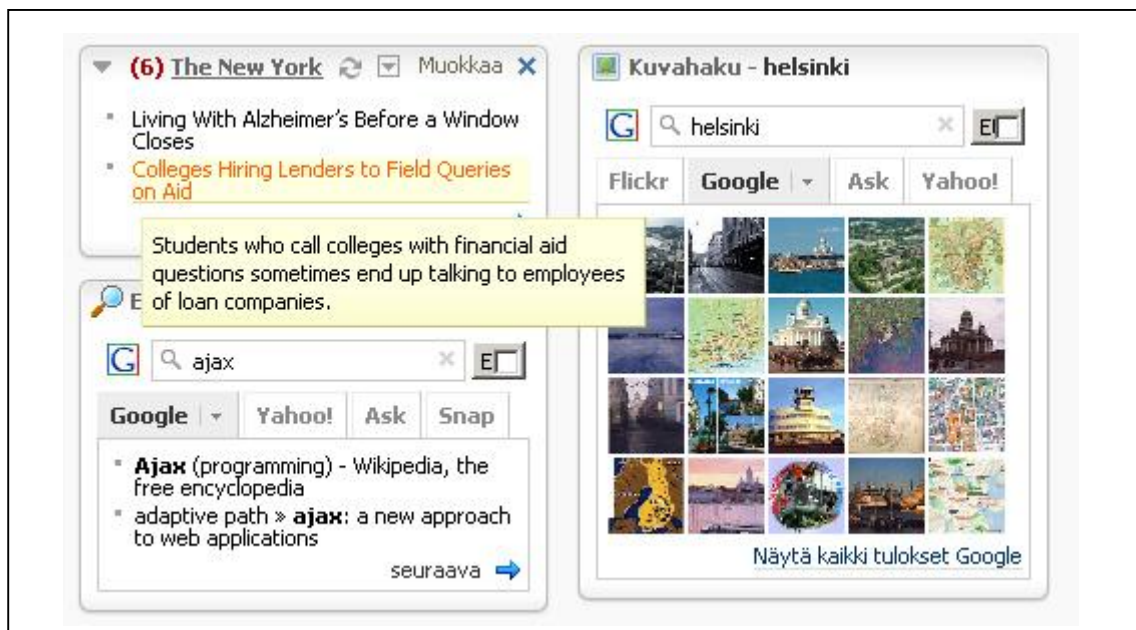


KUVA 11. Oasis-sivuston käyttämä kuvasuurennostekniikka (Oasis).

Perinteisissä sovelluksissa näytössä olevaa tilaa on maksimoitu mm. JavaScriptin avulla tehtyjen pudotusvalikkojen avulla. AJAX-sovelluksissa tilan maksimointi viedään vielä pidemmälle. AJAX mahdollistaa paljon laajempien kokonaisuuksien piilottamisen. Uutissivustoilla tätä hyödynnetään näyttämällä lukijalle esimerkiksi vain otsikot, joiden ingressi saadaan näkyviin, kun hiiri viedään otsikon päälle. Tämä voidaan toteuttaa

useammalla eri tavalla. Ingressi voi esimerkiksi työntää muita otsikoita alaspäin ilmestyessään otsikon alle tai se voidaan tuoda kelluvaan ikkunaan, joka aukeaa näytöllä olevien elementtien päälle.

Aloituspöytäaliksi kutsutaan sivustoa, josta käyttäjä voi muokata itselleen näköisensä aloitussivun. Näytölle voi raahata erilaista informaatiota sisältäviä muokattavia laatikoita. Halutessaan käyttäjä voi pienentää tai sulkea laatikot, vaihtaa niiden paikkaa tai muokata niiden ulkoasua ja sisältöä. Netvibes.com on toimiva esimerkki AJAX-tekniikoilla toteutetusta aloituspöytäalusta. Kuvassa 12 ruudulle on raahattu uutiset, jotka johtavat The New York Timesin sivuille, ja kaksi erilaista hakutoimintoa. Netvibes.com piilottaa tilan säästämiseksi uutisen ingressin kelluvaan ikkunaan, joka ilmestyy ruudulle onMouseOver-funktiolla. Sivusto päivittää uutiset automaattisesti tietyin väliajoin ja uusimmat otsikot noudetaan listaan RSS-syötteen avulla. Palvelu tallentaa kaikki tehdyt muutokset ja portaali pysyy muokatunlaisena seuraavalla sisäänkirjautumiskerralla.



KUVA 12. Netvibes.com-sivuston uutisen ingressi avautuu kelluvaan ikkunaan (Netvibes.com).

4.2 AJAXin heikkoudet

Ei ruusuja ilman risuja. AJAXilla on myös vastustajansa, jotka moittivat sen heikkouksia. AJAXin heikkoudet on listattu kuvassa 13.

SWOT-ANALYYSI

<p>VAHVUUDET</p> <ul style="list-style-type: none"> Nopeus Avoin standardi Läpinäkyvyys Tietojen tallennus Samankaltaisuus työpöydän ohjelmien kanssa 	<p>HEIKKOUEDET</p> <ul style="list-style-type: none"> Tietoturva Navigoinnin muuttuminen Yhteensopivuus Muistin varaaminen Trendikkyys
<p>MAHDOLLISUUDET</p> <ul style="list-style-type: none"> Leviäminen Kehittyminen 	<p>UHAT</p> <ul style="list-style-type: none"> Kilpailijat Trendit

KUVA 13. Heikkoudet.

4.2.1 Tietoturva

AJAX-sovelluksen suurimpana heikkoutena pidetään sen huonoa tietoturvaa. AJAX-sovelluksien hyökkäyspinnan (Attack Surface) väitetään oleva suurempi kuin perinteisen www-sovelluksen. Hyökkäyspinta on käsite, jolla mitataan hyökkäykselle auki olevia pisteitä. Ohjelmissa nämä pisteet ovat toimintoja, joihin kolmas osapuoli saattaa päästä käsiksi. Mitä pienempi pinta-ala on, sitä helpompi ohjelma on turvata. Väite AJAXin suuresta hyökkäyspinnasta ei kuitenkaan ole täysin paikkaansa pitävä. Sovelluksessa on toimintoja juuri sen verran, kun siihen on niitä koodattu. Käytetyllä tekniikalla ei siis ole yhteyttä hyökkäyspinnan suuruuteen. (Grossman 2006.)

Sovelluksissa esiintyvistä tietoturvaongelmista yleisimpiä ovat Cross-site request Forgery (XSRF) ja Cross-site scripting (XSS). XSRF ja XSS hyödyntävät auki olevia selainikkunoita kaapatakseen arkaluontoista informaatiota. Käyttäjä saattaa esimerkiksi

olla edelleen kirjautuneena pankkipalveluun, kun hän siirtyy vahingolliselle www-sivulle. Haitallisen sivun hiljainen AJAX-koodi voi tehdä pyynnön auki olevaan pankkipalveluun ja lähettää sieltä itselleen esimerkiksi käyttäjän sisäänkirjautumisen yhteydessä muodostuneita evästeitä. (Shah 2006, 3.) Tunnetuimmat asynkronista pyyntöä hyväksikäyttäneitä matoja ovat My Spacen kimppuun käynyt Samy Worm ja Yahooseen hyökännyt JS-Yamaner. Molemmat levisivät XMLHttpRequest-olion avulla. (Grossman 2006.)

Myös XML-dokumentteja pystytään vahingoittamaan erilaisin hyökkäyksin. Hyökkääjät voivat tuottaa vahingollisia XML-dokumentteja, jotka häiritsevät jäsentäjien palvelimella suorittamaa tulkintaa. (Shah 2006, 2.)

Hyökkäyksiltä voi suojautua ennen kaikkea hyvällä suunnittelulla. Suunnitteluun löytyy apua erilaisilta turvallisuutta käsitteleviltä sivuilta. Erinomainen on esimerkiksi Open Web Application Security Project. Suunnittelussa kannattaa huomioida mm. tiedon varmistaminen palvelin puolella, lomaketietojen validointi ja HTML-erikoismerkkien koodaaminen. Suunnitelmaan on hyvä sisällyttää myös jatkuvan heikkouksien etsiminen ja korjaaminen. (Frye 2006.)

4.2.2 Navigoinnin muuttuminen

AJAX on muuttanut liikkumisen verkossa aivan uudenlaiseksi. Enää ei navigoida sivulta toiselle, vaan sivut päivitetään dynaamisesti. Muutos toi mukanaan myös useita ongelmia.

Sivuston selaimessa sijaitseva historia-olio tallentaa kaikki sivut, jotka selaimelle on ladattu. Näillä sivuhistoriaan tallentuneilla sivuilla voi liikkua back-painikkeen avulla. AJAX ei kuitenkaan lataa uusia sivuja, vaan päivittää niitä dynaamisesti, eivätkä XMLHttpRequest-olion avulla tehdyt pyynnöt tallennu selaimen historiaan. (Holzner 2006, 325–326.) Sama ongelma havaittiin aikoinaan jo Flash-tekniikan yleistyessä ja siihen on kehitetty useita ratkaisuja, joita AJAX-sovelluksissa nyt sovelletaan. Sivuhistoriaa voidaan selata taaksepäin esimerkiksi iframen avulla, muokkaamalla URLia tai JavaScriptillä luodun oman back-painikkeen avulla. (Stenhouse 2005.)

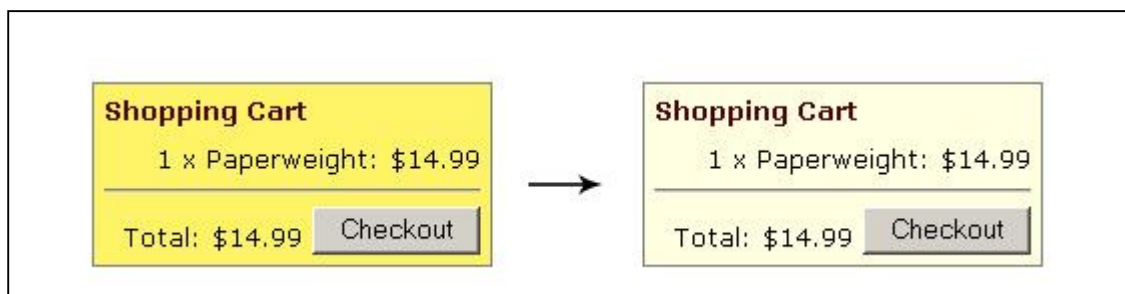
URL-ongelman vuoksi kirjanmerkit ja hakukoneet eivät toimi kunnolla. Käyttäjä ei voi tallentaa muokkaamiaan erillisiä sivuja suosikit-kansioon, sillä niillä kaikilla on sama URL-osoite. Kirjanmerkit korvataan AJAX-sovelluksissa usein itse rakennetuilla staattisilla linkeillä. Hakukoneet etsivät tietoa HREF- ja SRC-tageista, mutta eivät JavaScript-koodista. Ne eivät siis pääse kiinni asynkronisilla pyynnöillä tehtyihin sivun muutoksiin. Yksi ratkaisu tähän ongelmaan on esimerkiksi Googlen tarjoama Sitemaps-palvelu, jonka avulla voi kertoa hakukoneelle, mitä linkkejä se ei löydä, ja saada näin kyseiset linkit näkymään Googlen hakukoneessa. (Bosworth 2005.)

AJAX-sovelluksen läpinäkyvyys ja yllättävät muutokset voivat hämmentää tai ärsyttää käyttäjää. AJAX-sovellus saattaa vaihtaa osan sivun sisällöstä yllättäen tai tallentaa tietoa käyttäjän tahtomatta. Ja kuten luvun alussa kerrottiin, aiheuttaa läpinäkyvyys ärsytyksen lisäksi myös turvallisuusriskin. (Bosworth 2005.) Kolmas läpinäkyvyyden haittavaikutus on se, että pitkä latausaika saatetaan tulkita ohjelman toimimattomuutena. Tämän vuoksi latauksia on hyvä havainnollistaa erilaisin visuaalisin vinkein. Kuvassa 14 on esitetty kolme mahdollista tapaa havainnollistaa latautumista visuaalisesti. (Holzner 2006, 326.)



KUVA 14. Visuaalisia vinkkejä.

Basecamp käyttää sivuillaan havainnollistamiseen tyylikästä keltaista taustaväriä. Sivustolla käyttäjä voi esimerkiksi muokata lukemaansa uutista. Muokkauksen jälkeen ilmestyy uutisen taakse keltainen taustaväri. Väristä ei muodostu häiritsevää tekijää, sillä se haihtuu hiljalleen pois (ks. Basecamp). Samanlaista hiljalleen himmenevää keltaista huomioväriä käytetään myös Smarxwearin sivuilla ostoskorin yhteydessä. Kuvassa 15 ostoskoriin on lisätty yksi tuote. Lisätystä tuotteesta informoidaan keltaisen huomiovärin avulla. Väri haihtuu noin sekunnin kuluessa.



KUVA 15. Ostoskorin huomioväri (Smarxwear).

AJAX luo puitteet uusille monipuolisemmille käyttöliittymille. Valitettavan monissa käyttöliittymissä uusia ja tuntemattomia toimintoja ei kuitenkaan ole havainnollistettu tarpeeksi hyvin ja niiden käyttö vaatii opettelua. Käyttäjä saattaa oppia toiminnot yrityksen ja erehdyksen kautta, mutta opettelu vie kuitenkin aikaa, mitä ei voida pitää hyvänä käytettävyytenä. (Bosworth 2005.)

Monille sivuille mainokset ovat ainoa tulonlähde ja siksi niiden toimivuus on tärkeää. Perinteisessä WWW-sovelluksessa selain päivittyy joka kerta, kun sisältö vaihtuu eli siirrytään sivulta toiselle. Tätä kutsutaan mainosmaailmassa sivun lataukseksi (page view). Sivun latauksen yhteydessä myös sivun mainokset vaihtuvat. AJAX-sovelluksissa sivuja ladataan harvoin, minkä vuoksi sivun latauskerrat ovat huomattavasti alhaisempia kuin perinteisissä sovelluksissa. AJAX-sovelluksiin on kehitetty erilaisia tapoja mainosvaihtuvuuden lisäämiseksi. Yleinen tapa on päivittää sivu tietyn väliajoin, mutta se ei ole suositeltavaa. (Picard 2006.) Toinen tapa on lähettää mainospyyntö palvelimelle aina kun sivun sisältö muuttuu. Pyyntö palauttaa kehyksen (iframe) sisälle uuden mainoksen. (Cho 2006.) Vaihtoehtoinen hinnoittelutapa sivun lataukselle on CPC (Cost per Click). Mainostilasta maksetaan klikkauksien määrän mukaan. Eli mitä enemmän mainosta klikataan, sitä enemmän mainostaja siitä maksaa. (Google AdSense, 2007.)

4.2.3 Yhteensopivuus

AJAX-ohjelmat koodataan JavaScriptillä, mikä saattaa aiheuttaa ongelmia selaimien kanssa. On olemassa selaimia, joissa on mahdollista estää JavaScriptin käyttö eli ns. sulkea JavaScript pois käytöstä. (Holzner 2006, 328.) W3C:n JavaScript-tilaston mukaan tammikuussa 2007 6 % käyttäjistä oli sulkenut JavaScriptin pois päältä (W3C 2007). Sovelluksien on oltava esteettömiä eli niiden on toimittava myös ilman

JavaScriptiä. Esimerkiksi Yagoon uutissivusto toimii erinomaisesti, vaikka JavaScript kytkettäisiin pois päältä. Käytettävyyttä on parannettu AJAXilla, mutta pohjalta löytyy kuitenkin perinteinen HTML-sivusto.

Selainten kanssa ongelmia aiheuttaa myös niiden erilaiset tavat JavaScriptin tapahtumakäsittelylle. Internet Explorer käyttää selaimissaan Active X-oliota ja muut selaimet XMLHttpRequest-oliota. Http-kutsu on tämän vuoksi tehtävä niin, että IE:stä tehdyt kutsut luovat Active X-olion ja muista selaimista tehdyt kutsut luovat XMLHttpRequestin. (Bittiavaruus.)

Yhteensopivuus ongelmia voi ilmetä myös eri yritysten tekemien AJAX-kirjastojen ja selainten välillä. Osa kirjastoista toimii ainoastaan Internet Explorerissa ja suurin osa vain Explorerissa ja Firefoxissa. Monet kirjastot eivät tue esimerkiksi Safaria ollenkaan, vaikka se on ensimmäinen selain, joka läpäisi Web Standards -projektin Acid 2 -testin. Testin avulla tutkitaan, kuinka hyvin selain tukee verkkostandardeja. (Scott, 2006.)

4.2.4 Muistin varaaminen

Rikkaat käyttöliittymät vaativat mutkikkaita ohjelmia. Monipuolisen toiminnallisuuden vuoksi AJAX-sovellukset saattavat olla suuria ja varata paljon muistitilaa. Sovelluksen toteutusvaiheessa kannattaa poistaa suuret JavaScript-oliot, jotka kuormittavat muistia liikaa. Olioiden liian suuri koko voidaan välttää myös suunnittelemalla ohjelmat tarkkaan ja kehittämällä niiden koodia. (Holzner 2006, 328.)

Koodin määrää voidaan kontrolloida kehyksien avulla. Kehysten avulla koodimäärästä tulee pienempi, koodi on parempaa ja syntyy vähemmän selainbugeja. Kehykset käyttävät usein On-Demand-komentoa, jonka avulla ladataan sivulle vain tarvittava JavaScript-koodi. Näin ollen sivustoa ei kuormiteta tarpeettomalla koodilla. (Ajax Patterns 2007.)

4.2.5 Trendikkyys

Tällä hetkellä AJAX on ”se juttu” verkkosovellusten toteuttamiseksi. Se on malli, jota kaikki trendikkäät yritykset käyttävät sivuillaan. AJAXia ei pidä kuitenkaan käyttää vain sen takia, että se on uusi ja trendikäs. Sitä tulisi käyttää vain ja ainoastaan silloin, kun se on tarpeellista ja kun koodaaja tuntee kyseiset tekniikat ja osaa käyttää niitä. Trendin innoittamina AJAX-sovellukset sortuvat usein liialliseen dynaamisuuteen ja sekavuuteen. Sovellus kannattaa pitää selkeänä ja yksinkertaisena ja välttää turhan luovia ratkaisuja. (Crane & al. 2006, 212–216.) Sivuston on toimittava myös ilman AJAX-toimintoja.

Klassinen esimerkki trendikkästä, mutta huonosta AJAX-sovelluksesta on Zago LLC:n sivut (ks. kuva 16). Sivusto on ulkoasultaan tyylikäs, mutta sen ongelmalista on valitettavan pitkä. Eniten häiritsee etusivun ihmeellinen ja yllätyksellinen vilkkuminen, mikä johtuu ajastetusta XMLHttpRequest-pyynnöstä. Olio hakee tietyn väliajoin palvelimelta sivulle uuden taustakuvan ja värimaailman. Sivuhistorian toiminta on puoliksi korjattu, mutta kirjanmerkit eivät edelleen toimi. Back-painike toimii tasojen (esim. what we do) sisällä, mutta tasosta toiseen sen kanssa ei voi navigoida.



KUVA 16. Zago LLC:n sivuston ulkoasu (Zago LLC).

4.3 AJAXin mahdollisuudet ja uhat

Tulevaisuudessa AJAXin on mahdollista levittäytyä laajempaan käyttöön ja kehittyä vieläkin paremmaksi malliksi. AJAX joutuu kohtaamaan myös erilaisia uhkia, joista näkyvimpiä ovat itse mallin, sen käyttämien tekniikoiden ja työkalujen kilpailijat. Kuvassa 17 on listattu AJAX-sovellusten mahdollisuudet ja uhat.

SWOT-ANALYYSI

<p>VAHVUUDET</p> <ul style="list-style-type: none"> Nopeus Avoin standardi Läpinäkyvyys Tietojen tallennus Samankaltaisuus työpöydän ohjelmien kanssa 	<p>HEIKKOUEDET</p> <ul style="list-style-type: none"> Tietoturva Navigoinnin muuttuminen Yhteensopivuus Muistin varaaminen Trendikkyys
<p>MAHDOLLISUUDET</p> <ul style="list-style-type: none"> Leviäminen Kehittyminen 	<p>UHAT</p> <ul style="list-style-type: none"> Kilpailijat Trendit

KUVA 17. Mahdollisuudet ja uhat.

4.3.1 Mahdollisuudet tulevaisuudessa

AJAX on jo tällä hetkellä suosittu malli verkkosovellusten tuottamiseen, mutta sillä on mahdollisuus vielä laajempaan levinneisyyteen. SitePointin ja Ektronin tekemän State of Web Development 2006/2007 -tutkimuksen mukaan kesällä 2006 käytetyin tekniikka dynaamisten WWW-sivujen tuottamiseen oli Adoben Flash. 5000 verkkokehittäjästä 39 % käytti projekteissaan Flashia ja 30 % käytti AJAXia. Tutkimuksen mukaan vuoden 2007 kesään mennessä 45,7 % WWW-kehittäjistä aikoo ottaa AJAX-tekniikat käyttöönsä ja 27,36 % aikoo vuorostaan ottaa Flashin käyttöönsä. Mikäli nämä ennusteet pitävät paikkansa, on AJAX kesällä 2007 suosituampi kuin Flash. (Aponovich et al. 2006.)

AJAX-lähtöisten verkkosovellusten nopean leviämisen asialla on myös Open Ajax Alliance. Open Ajax -yhteisö koostuu markkinajohtajista, open source -projekteista ja yrityksistä. Ryhmä ratkoo AJAXin yhteensopivuusongelmia ja opastaa käyttäjäyhteisöä AJAXin onnistuneeseen käyttöön. Tavoitteena on saattaa AJAX suuren massan tietoisuuteen ja käyttöön. (Learningbusiness.fi 2006.)

AJAX hyötyisi laajasta levinneisyydestä eniten tekniikoiden kehittymisen nopeutumisenä. Mitä useampi käyttäjä AJAXilla on, sitä suuremmalla yleisöllä olisi intressi kehittää sen tekniikoita. Myös yritykset olisivat innokkaampia kehittämään ja maksamaan AJAX-sovelluksista erottuakseen massasta ja pysyäksään kehityksessä mukana. Yritykset olisivat myös valmiita panostamaan AJAX-ongelmien ratkaisemiseen. Laaja levinneisyys edesauttaa myös standardien paranemista.

Vaikka AJAX on suosittu, sillä on vielä kehittymisen varaa. Aiemmin esitetystä SWOT-analyysistä huomaa, että heikkouksien lista on vielä pitkä. Toki heikkouksia on myös kaikilla AJAXin kilpailijoilla, mutta pysyäksään kehityksessä mukana, on AJAXin korjattava heikkoutensa. Tätä prosessia haluaa nopeuttaa myös aiemmin mainittu Open Ajax Alliance.

Työpöydän ohjelmia siirretään jatkuvasti verkkoon, mutta monipuolisimpien ohjelmien verkkototeutuksiin ei tekniikka vielä riitä. Tämän mahdollistamiseksi on JavaScript-ohjelmia kehitettävä. Merkkejä monimutkaisempien työpöydän ohjelmien siirtämisestä verkkoon on jo näkyvissä. Huhun mukaan esimerkiksi Google olisi kehittämässä omaa Google Officea, jonka toimintaperiaate olisi hyvin samankaltainen kuin MS Officen. Adobe valmistele puolestaan tunnetun kuvankäsittelyohjelmansa, Photoshopin, siirtämistä verkkoon. Verkkoympäristö luo ohjelmien siirtoon omat ongelmansa. Esimerkiksi mahdollisesta Googlen Officesta olisi hyvä muokata businesskäyttöön versio, joka toimii palomuurin takana. (Mahemoff 2006, 1.)

Yksi kehityksen suunta on myös AJAXin yhteiskäyttö muiden mallien ja tekniikoiden kanssa. Tällä hetkellä esimerkiksi Microsoftin .NET:llä, Flexillä ja OpenLaszolla on AJAX-tuki. Yhdistelemällä tekniikoita voidaan hyödyntää niiden vahvuuksia ja optimoida sivuston käytettävyys. (Mahemoff 2006, 1.)

4.3.2 Tulevaisuuden uhat

AJAXin pahin uhka on ehdottomasti sen kilpailijat. Rikkaita verkkosovelluksia voidaan tuottaa usealla eri tekniikalla ja käyttäen useita eri kehyksiä ja alustoja. Kilpakentälle ilmestyy tiuhaan tahtiin myös uusia tekniikoita ja malleja, jotka uhkaavat nykyisten tekniikoiden asemaa. AJAX on itse hyvä esimerkki uuden mallin ilmestymisestä verkko-ohjelmoinnin kentälle.

Eniten AJAXia verrataan varmasti Flashiin. Flash on Macromedian kehittämä (nykyään Adobe'n omistuksessa) integroitunut kehitysympäristö. Termillä viitataan usein myös Flash Playeriin, joka on Flash-animaatioiden pyörittämiseen tarkoitettu käyttäjäpuolen sovellus. Flash-sovellusten ohjelmointikielenä käytetään JavaScript-kielen lähisukulaista, Action-Scriptiä. Flashin vahvuus on sen AJAXia huomattavasti parempi tuki vektorigrafiikalle, animaatiolle ja äänille. AJAX-sovelluksiin verrattuna Flash-sovellusten latausajat ovat pidempiä ja AJAX hakkaakin Flashin mennessä nopeiden ja ruuhkaisten sivustojen toteuttajana. Ei olekaan mikään ihme, että Google on valinnut tekniikakseen AJAXin. (Pasztory.)

Adobe on kehittänyt aikajanaan perustuvan Flashin rinnalle ohjelmointiperusteisen Flexin, joka soveltuu paremmin internet-sovellusten käyttöliittymän kehittämiseen. Adobe'n Flex on tekniikkaryhmä rikkaiden verkkosovellusten tekemiseen. Se käyttää merkintäkielenä XML-kieleen perustuvaa MXML-kieltä ja ohjelmointikielenä Action Scriptiä. Flexissä on erinomainen graafinen käyttöliittymä, joka helpottaa ja nopeuttaa sovellusten tuottamista. Ensimmäinen Flex-versio ei saavuttanut vielä suurta suosiota kalliin hintansa vuoksi. Tammikuussa 2007 Adobe julkaisi muillekin kuin suuryrityksille tarkoitetun Flex 2 SDK:n, joka on ilmainen. (Wikipedia 2007 Adobe Flex.)

Adobe on julkaissut myös beta-version Adobe Apollosta, joka julkaistaan loppuvuodesta 2007. Se on ajoympäristö, jossa voi toteuttaa työpöydän ohjelmia verkkotekniikoilla. Ohjelmat toimivat erinomaisesti myös offline-tilassa. Apollon AJAX-tuki on hyvä, mutta koska se on kehitetty Flashin rinnalle, vaatii se Flash Playerin lataamisen. (Adobe Labs 2006.)

Varteen otettava kilpailija verkkosovellusten maailmassa on myös OpenLaszlo, joka on ilmainen open source -alusta. OpenLaszlo on kirjoitettu XML-kieleen perustuvalla LZX-merkintäkielellä ja JavaScriptillä, eikä se vaadi kehitysympäristön asennusta. OpenLaszlo voidaan kääntää Flashiksi ja DHTML:ksi. (Laszlo Systems Inc. 2006.)

Edellä mainittujen tekniikoiden ja mallien lisäksi myös Java-pohjaiset sovellukset kilpailevat WWW-kehittäjien suosiosta. Java on teknologiaperhe, johon kuuluu ohjelmointikielen lisäksi ajoaikainen ympäristö. Selaimessa toteutettavia appletteja voidaan käyttää latauksen jälkeen myös ilman internet-yhteyttä, sillä vuorovaikutusta palvelimen kanssa ei enää tarvita. Appletit ajetaan eristetyssä hiekkalaatikossa, minkä vuoksi niiden tietoturvariski on huomattavasti JavaScript-ohjelmia pienempi. (Crane & al. 2006, 28–29.) AJAX-sovelluksiin verrattuna myös applettien selaintuki on parempi. JavaScriptin etuna on puolestaan se, ettei se vaadi minkäänlaisen alustan tai ohjelman asennusta. AJAX-sovellukset ovat myös kevyempiä ja nopeampia kuin Java-Appletit. (Eckel 2007.)

Microsoft on myös julkaissut oman järjestelmänsä rikkaiden verkkosovellusten tuottamiseksi. Windows Presentation Foundation (WPF) toimii uuden Windows Vista-käyttöliittymän pohjana ja ohjelmoitavana rajapintana. Se käyttää XAML-merkintäkieltä, .NET-ohjelmointikieltä ja vektorigrafiikkaa. WPF-sovellukset voidaan sijoittaa työpöydälle tai selaimen XAML-selainsovelluksina (XBAP). Sovellukset hyödyntävät näytönohjaimen 3D-ominaisuuksia. WPF toimii erinomaisesti Microsoftin ympäristössä, mutta vaatii muihin kuin IE-selaimiin liitännäisen. WPF ei toimi Macintoshin tietokoneissa. Microsoft haastaa AJAXin myös Atlas-kehityksensä kanssa. Atlas käyttää JavaScriptin tilalla ohjelmointikielenä C#-kieltä. (Wikipedia 2007 Windows Presentation Foundation.)

Mozilla on kehittänyt rikkaiden verkkosovellusten tuottamiseen XUL-kuvauskielen. Sen ongelmana on se, ettei se toimi muissa kuin Mozillan selaimissa. Potentiaalia XUL-sovelluksilla riittää, mutta Mozillan on ensin ratkottava sovellusten yhteensopivuusongelma muihin selaimiin. XUL-sovellukset toimivat myös työpöydällä offline-tilassa. (The Mozilla Organization 2007.)

Rikkaiden verkkosovellusten tekemiseen käytetään useita erilaisia tekniikoita ja malleja, kuten aiemmin mainituista AJAXin kilpailijoista huomaa. Vertailu niiden kesken ei ole aivan yksioikoista. Ongelma muodostuu jo siitä, että AJAX on käsitteenä niin hutera. Se on tekniikkaryhmä, malli tai ajatusmalli, kun taas sen kilpailijat ovat selvästi kehyksiä, ympäristöjä, ohjelmointikieliä tai alustoja. AJAXin kilpailijoita on verrattu tarkemmin liitteessä 2.

Verkkosovellusten teossa käytetään usein myös tekniikoita, malleja, kehyksiä, alustoja ja ympäristöjä sekaisin. AJAX voidaan toteuttaa puhtaasti yksinkertaisessa tekstieditorissa, mutta tehokkaammin sitä voi käyttää erilaisissa kehyksissä. AJAXille on tehty paljon omia kehyksiä, mutta sitä voidaan käyttää myös esimerkiksi Flexissä. AJAXiin voidaan lisätä myös ulkopuolisia tekniikoita. Esimerkiksi SVG-kieltä lisäämällä saavutetaan Flashin vektorigrafiikan taso. RIA-menetelmien välinen kilpailu on kuitenkin tärkeää, koska se edistää tekniikoiden kehittämistä. Hyvä esimerkki tästä on Internet Explorer -selaimen kehityksen pysähtyminen, kun se saavutti ylivoimaisen suosion muihin selaimiin verrattuna. Sitä alettiin kehittää uudestaan vasta, kun Mozillan selain alkoi olla sille uhka.

Käytettävä tekniikka on usein riippuvainen sovelluksen tarkoituksesta. Suunnitteluvaiheessa on tärkeä miettiä, mikä olisi järkevin tapa toteuttaa verkkosovellus. Peliä ei kannata lähteä toteuttamaan tekniikalla, joka on parhaimmillaan kevyissä palveluissa. Taulukossa 4 on listattu eri menetelmille parhaiten soveltuvia käytön kohteita.

Taulukko 4. Verkon kehitykseen käytettävät tekniikat ja niiden käyttötarkoitus.

Tekniikka	Soveltuu parhaiten
AJAX	Pienet päivitykset, ruuhkaiset palvelut, tekstisisältöiset sivut
Flash	Animaatiot ja videot
Java Applet	Ohjelmistojen toimintoja vaativat sovellukset
WPF (XBAP)	Näyttävät visuaaliset toteutukset
OpenLaszlo	Animaatiot, tekstisisältöiset sivut
Flex	Suuret bisnespalvelut

IT-ala on nopeasti elävää ja trendit vaihtuvat tiheään. AJAX on varsin uusi tapa toteuttaa verkkosovelluksia, mutta on mahdollista, että sen tilalle löydetään jokin vielä tehokkaampi ja parempi tapa. AJAX mahdollinen korvaaja olisi interaktiivisempi, laajemmin levinyt ja mahdollistaisi Java-tyyppisen monialustaisen kehityksen. Tätä voisi kutsua AJAX 2.0:ksi. (No Relation 2007.)

4.4 AJAXin turvattu tulevaisuus

AJAX on tällä hetkellä suosittu kuin koskaan. AJAX on niin suosittu, että sen suosion pelätään romahtavan hätäisesti tehtyjen huonojen sovellusten vuoksi. Jääkö AJAX hetken trendiksi vai onko sillä potentiaalia selvitä muiden verkkotekniikoiden rinnalla?

AJAX pisti monipuolisilla toiminnoillaan verkkoajattelun uusiksi. WWW-kehittäjien täytyy tämän vuoksi ottaa entistä enemmän asioita huomioon suunnitteluvaiheessa, mikä on hidastanut AJAXin leviämistä. Vuorovaikutus on entistä rikkaampaa, mutta myös monimutkaisempaa hallita. Tämän uuden ajattelumallin vuoksi AJAXia ei ole vielä omaksuttu kunnolla yritysmaailmassa, ei ainakaan Suomessa. Ohjelmoijat hallitsevat varmasti jo tarvittavat tekniikat, mutta yritykset eivät vielä osaa vaatia uusia käytettävyydeltään parempia muotoja sivuilleen. AJAX tekee kuitenkin tuloaan kotimaisten yritysten verkkosivujen mallina. Esimerkiksi Fonectan 020202-palvelu on täysin AJAX-tekniikoilla koodattu sovellus.

WWW-kehittäjien ja yritysten hitaaseen AJAX-syttymiseen on ollut osasyynä myös hyvien työkalujen ja kirjastojen vähyys. Uudet markkinoille ilmestyneet kehykset ja työkalut ovat hyviä ja niitä julkaistaan koko ajan lisää. Markkinajohtajat ovat havainneet AJAXin ja ennen kaikkea rikkaiden verkkosovellusten maailmanlaajuisen suosion, minkä vuoksi yritykset kehittävät vauhdilla omia kehyksiä ja AJAX-kirjastoja markkinoille. AJAXin kannalta on tärkeää saada markkinoille nimenomaan AJAXille suunnattuja kehyksiä.

Hyvien kehitysien ja kirjastojen lisäksi AJAXin asema paranee myös selainten kehittyessä. Todennäköistä on, että sekä Microsoft että Mozilla lisäävät selaintensa AJAX-tukea. Tulevaisuudessa parannusta odotetaan ainakin cross site scripting - ongelmaan, turvallisuuteen, widgetien ja JavaScriptin tukemiseen. (Kappe 2007.)

AJAXin tulevaisuus näyttää turvatulta. Yksi merkki tästä on se, että Microsoft ja Google liittyivät Open Ajax Allianceen 20.3.2007. Microsoft mietti jäsenyyttään kauan, koska yhtiö julkaisi äskettäin oman ratkaisunsa dynaamisten WWW-sivujen tuottamiseksi, eikä ollut halukas antamaan AJAX-kirjastojaan yhteisön käyttöön. Keskustelupalstoilla yhtiö sai ennen liittymistään negatiivista palautetta, siitä ettei se edistä avoimen koodin kehittämistä. Nyt suurimmat IT-alan yritykset kuuluvat yhteisöön ja näin ollen AJAXia voidaan kehittää enemmän yhtenäisen standardin mukaisesti.

AJAXin asemaa on selvitetty myös tutkimuksilla. Luvussa 4 mainittu State of Web Development -tutkimus ennusti AJAXin olevan käytetyin verkkosovellusten tekemiseksi tarkoitettu tekniikka kesällä 2007. Sitä AJAX varmasti tulee olemaan. Tutkimuksen perusteella ei kuitenkaan käy ilmi käytetäänkö verkkosovellusten tekemiseen ainoastaan AJAXia vai käytetäänkö sen kanssa myös muita tekniikoita, kuten Flashia. (Aponovich et al. 2006.) Tulevaisuudessa kehitys on menossa siihen suuntaan, että tekniikoita käytetään yhdessä paremman käytettävyyden toteuttamiseksi.

Ei siis voida sanoa, että jokin kilpailija olisi suoranaisesti syrjäyttämässä AJAXia. Todennäköistä on, että hehkutus AJAXin ympärillä hiljenee ja AJAX siirtyy CSS:n kaltaiseksi taustavaikuttajaksi. Sitä tullaan käyttämään yhdessä muiden tekniikoiden, kehitysien ja alustojen kanssa.

Flashin ja AJAXin yhteiskäyttöä esiintyy jo nyt internetissä. Esimerkiksi Backbase on rakentanut dynaamisen pankkikonttorien paikantimen, jonka perustoiminnot on toteutettu AJAX-tekniikoilla. Flashia on puolestaan käytetty animaation luomiseen. Kuvassa 18 Flashilla on toteutettu kartta, jota voi zoomata ja liikuttaa.



KUVA 18. Pankkikonttorien paikannuspalvelun Flash-kartta (Backbase).

Oma kilpailunsa tullaan käymään AJAX-kehittäjien suosiosta kisaavien kehysten välillä. Microsoft julkaisi tammikuussa 2007 uuden Vista-käyttöjärjestelmän ja sen mukana WPF:n, jonka avulla AJAX-gurut voivat lisätä sivuilleen interaktiivisuutta ja uusia piirteitä. Adobe julkaisee vuoden 2007 lopulla Apollon, joka mahdollistaa ohjelmien teon työpöydälle. Apollo tulee nopeuttamaan myös edellä mainittua AJAXin ja Flashin yhteiskäyttöä sovelluksissa. Avoimen koodin kannattajille on puolestaan tarjolla maaliskuussa 2007 julkaistu OpenLaszlo 4.0. (Stewart 2006.)

AJAXin käyttämien tekniikoiden lisäksi WWW-kehittäjät saattavat kiinnittää huomionsa myös muihin jo olemassa oleviin tekniikoihin. Näitä tekniikoita käytetään jo joissain sovelluksissa, mutta ne eivät ole vielä saaneet tulta alleen. Tällaisia tekniikoita ovat esimerkiksi http Streaming (Comet), Cross-Domain JSON, Unique URLs ja Lazy Registration. (Mahenoff 2006.) Nämä tekniikat tuskin korvaavat AJAXia, vaan ennemminkin tukevat sitä. JSON:a käytetään jo nyt jossain määrin XML:n tilalla AJAX-sovellusten metakielenä.

Tällä hetkellä ilmapiiri AJAXin ympärillä vaikuttaa positiiviselta, eikä sitä olla hylkäämässä taka-alalle. Laajan suosionsa ansiosta se tulee säilyttämään hyvän aseman muiden verkkotekniikoiden joukossa. Se miten hyvästä asemasta on kyse, selviää seuraavien parin vuoden kuluessa.

5 YHTEENVETO

Web 2.0:n siivellä suureen suosioon on noussut AJAX. Se on ryhmä tekniikoita, joilla toteutetaan dynaamisia ja rikkaita verkkosovelluksia. AJAXissa käytetyt tekniikat ovat entuudestaan tuttuja, mutta niiden nykyinen käyttö tuli mahdolliseksi vasta tekniikan kehityttyä tarpeeksi.

AJAX-sovelluksen ulkoasu koodataan vanhalla tutulla HTML:llä tai tarkempia määräyksiä noudattavalla XHTML:llä sekä tyyliä määrittävällä CSS:llä. Näiden avulla saadaan aikaiseksi katsojan näkemä WWW-sivu. Sivujen toiminnallisuus koodataan JavaScriptillä DOM-solmupuuta hyväksikäyttäen. JavaScriptin avulla koodataan AJAXin sydän eli XMLHttpRequest-olio, joka tekee palvelimelle asynkronisia pyyntöjä. Asynkronisella tarkoitetaan sitä, ettei selain pysähdy odottamaan palvelimen vastausta, vaan jatkaa toimintaansa normaalisti. Palvelimen ja asiakaspuolen välillä liikkuva data kirjoitetaan yleensä XML-muotoon.

AJAXin valtaisa suosio on pitkälti XMLHttpRequestin ansiota tai oikeastaan sen, että Google huomasi oliolla saavutettavat edut ja otti AJAXin käyttöönsä.

XMLHttpRequest-olion avulla käytettävyys sai uusia ulottuvuuksia ja perinteinen navigointitapa muokattiin uuteen uskoon. Enää ei tarvinnut päivittää koko sivua tai siirtää suuria määriä dataa, minkä vuoksi käyttäjän ja sovelluksen välinen interaktio nopeutui huomattavasti.

Asynkronisuuden ja nopeuden lisäksi AJAXin vahvuuksia ovat kyky tallentaa tietoa ja mahdollisuus muokata elementtien sijaintia www-sivulla. Internetistä löytyy hienoja sivustoja, joiden toiminnallisuus on työpöydän ohjelmien tasolla. Netvibes.comissa voi muokata aloitusportaalikseen haluamansa näköisen ja sisältöisen informaatiopakettin, josta löytyy lähes kaikki tarvittava uutisista sähköpostiin ja blogiviesteihin. Sivusto on muokattavissa drag & drop -tekniikalla ja tehdyt muutokset tallentuvat automaattisesti – kätevää, käytettävää ja jotain mitä ei verkossa ole aiemmin nähty.

AJAXin vahvuudet tulevat parhaiten esille ruuhkaisilla sivustoilla, joiden sisältö on pääperiaatteeltaan tekstiä. Tunnetuimmat esimerkit ruuhkaisista tekstipohjaisista sivustoista ovat Googlen hakukone Google Suggest ja sähköpostipalvelu Gmail. Toki AJAX toimii myös runsaasti kuvia sisältävillä sivuilla, kuten Flickr, mutta raskaita visuaalisia kokonaisuuksia ei AJAXilla kannata lähteä toteuttamaan.

Mikään tekniikka tai malli ei ole kuitenkaan täydellinen. AJAXilla on heikkoutensa ja äänekkäät vastustajansa. Nopeinta reagointia kaipaa AJAXin turvallisuusriskien korjaaminen. XMLHttpRequest mahdollistaa läpinäkyvien pyyntöjen lähettämisen käyttäjän huomaamatta. Haitalliset sivustot saattavat hyödyntää sitä hyökätessään muille sivustoille. Ne lähettävät esimerkiksi pankkipalveluille pyyntöjä pankin asiakkaan nimissä pyrkien saamaan tietoonsa asiakkaan kirjautumisen yhteydessä muodostuneita evästeitä. Haittaohjelmat ovat jo hyökänneet mm. Yahooon ja My Spacen kaltaisten jättien kimppuun.

Navigoinnin muuttuminen toi mukanaan myös käyttäjiä ärsyttäviä piirteitä. Tavalliset käyttäjät haikailevat back-painikkeiden ja kirjanmerkkien perään ja yritykset ovat huolissaan mainonnan ja sen seurannan muuttumisesta. Back-painikkeen ja kirjanmerkkien toiminta perustuu sivuhistorian tallentamiin WWW-sivuihin, joilla käyttäjä vierailee surffatessaan verkossa. AJAX-sovelluksessa ei kuitenkaan siirrytä sivulta toiselle perinteiseen tapaan, vaan sivu päivitetään dynaamisesti. Sama ongelma koskettaa myös mainontaa, sillä mainokset ovat vaihtuneet aina käyttäjän siirtyessä uudelle sivulle. Vanhalla systeemillä sama mainos saattaisi pysyä esillä koko ajan.

AJAX-sovellusten ongelmiin on kehitetty jo monia ratkaisuja, joista osa on toimivia ja osa valitettavan monimutkaisia. Samoja ongelmia on esiintynyt myös AJAXin kilpailijoilla, kuten Flashilla. AJAXin levitessä yhä laajemmalle nopeutuu myös sen ongelmanratkaisu. WWW-kehittäjät kehittävät skripteistä nopeampi ja toimivampia ja hyödyntävät JavaScript-ohjelmia moninaisemmilla tavoilla. Tähän mennessä on JavaScript-ohjelmilla ratkottu mm. back-painike-ongelma ohjelmoimalla sen toiminta itse.

Skriptien kehittyessä yhä useampi työpöydän sovellus pystytään tuomaan verkkoon. Verkkosovelluksina verkossa on jo hyviä ja toimivia versioita mm. tekstinkäsittelyohjelmista ja Outlook-sähköpostiohjelmasta. Monimutkaisemmat ohjelmat tekevät myös jo tuloaan, sillä Adoben huhutaan tuovan Photoshop-kuvankäsittelyohjelma verkkoon ja Googlen ennustetaan tuottavan oman Office-paketin verkkokäyttöön.

Google on AJAXin vankka kannattaja, mutta Adobesta ei samaa voi sanoa. Adoben tallista löytyy AJAXin pahimpana pidetty kilpailija Flash ja sen ohjelmoijille suunnattu versio Flex, josta tosin löytyy myös AJAX-tuki. AJAX on syrjäyttämässä Flashin dynaamisten WWW-sivujen mallina, mutta Flash vetää edelleen pidemmän korren animaatioiden, vektorigrafiikan ja videoiden saralla.

Markkinajohtajista myös Microsoft on tullut mukaan rikkaiden verkkosovellusten tuottamiseen. Yhtiö julkaisi tammikuussa uuden Vista-käyttöjärjestelmän ja sen siivellä näyttävien visuaalisten sovellusten tekoon tarkoitetun WPF:n. Microsoftin ongelmana on kuitenkin usein selainsopivuus. Sen omat sovellukset toimivat parhaiten Internet Explorerissa, mutta vaativat turhan usein liitännäisiä muissa selaimissa. Microsoft on aiheuttanut närää myös AJAX-piireissä, sillä sen selaimet eivät tue XMLHttpRequest-oliota, vaan se pitää kääntää Active X:ksi.

Ohjelmoijien suosiosta kilpailtaessa yksi AJAXin valttikorteista on sen avoimuus. AJAX perustuu avoimiin standardeihin, joita kaikki voivat kehittää. Flash ja WPF kuuluvat molemmat yhden omistajan alle, eikä niitä pääse muokkaamaan. AJAX ei kuitenkaan ole ainoa avoin verkkosovellusten tekoon tarkoitettu malli. Myös OpenLaszlo on avoin malli. Se kirjoitetaan XML-kieleen perustuvalla LZK-merkintäkielellä ja JavaScriptillä ja on käännettävissä Flashiksi tai DHTML:ksi.

Myös AJAX-tekniikoilla on kilpailijansa. JavaScript kilpailee suosiosta muiden ohjelmointikielten kanssa. JavaScript pohjautuu osittain Java-kieleen, joka on sen pahin uhka. Javalla voidaan toteuttaa monimutkaisempia ja raskaampia ohjelmia kuin JavaScriptillä. Javan appletit eivät kuitenkaan ole saavuttaneet suurta suosiota, koska ne vaativat Java-liitännäisen asentamisen. Java on tunkemassa AJAX-sovellusten ohjelmointikieleksi, mutta vaatii ainakin vielä työkalun (esim. Google Webtoolkit), jolla koodi käännetään JavaScriptiksi. Microsoftin Atlas haastaa JavaScriptin C#-kielellä.

Verkkosovellusten kehittämiseen käytetään yhä enenevässä määrin eri tekniikoita sekaisin. Tämän vuoksi ei voida vetää tarkkaa rajaa kilpailijoiden välille. Tekniikoita sekoitetaan siinä määrin, että ei ole selvää, voidaanko edes puhua mistään kilpailijoista. Onko Flash AJAXin kilpailija, vaikka niitä voidaan käyttää keskenään toimivien ja monipuolisten sovellusten tekemiseen. Entäpä WPF? Se kilpailee Flashin kanssa animaatioiden ja visuaalisesti näyttävien sovellusten tekemisessä ja AJAXin kanssa dynaamisten sovellusten tekemisessä. WPF:llä on kuitenkin AJAX-tuki ja niin on myös Flexillä ja OpenLaszolla.

Onko sillä loppujen lopuksi mitään merkitystä ovatko tekniikat toistensa kilpailijoita? On, koska kilpailu nopeuttaa tekniikan kehittymistä. Pysyäkseen kilpailussa mukana on mallien ja tekniikoiden kehityttävä muiden mukana. Hyvä esimerkki tästä on Internet Explorerin kehityksen pysähtyminen, kun siitä tuli laajimmin käytetty selain. Sitä alettiin toden teolla kehittää uudestaan vasta, kun Mozillan Firefox alkoi vallata jalan sijaa selainsuosiossa.

Tekniikoiden kehitys ja leviäminen edistää myös standardien muodostamista. Standardit parantavat sovelluksien saatavuutta pienentäen yhteensopivuusongelmia. Kehityksen edistämiseksi on perustettu erilaisia yhteisöjä, joiden tavoitteena on ratkoa ongelmia ja opastaa tekniikoiden oikeanlaisessa käytössä. AJAXin ympärille on esimerkiksi perustettu Open Ajax Alliance, jonka tavoitteena on neuvoa yrityksiä ja WWW-kehittäjiä AJAXin optimaaliseen käyttöön ja edistää ongelmanratkaisua erilaisten tiimien avulla.

Tämän opinnäytteen tutkimuksen perusteella AJAX ei ole häviämässä verkkosovellusten maailmasta. Se leviää edelleen ja sitä käytetään tulevaisuudessa enenevässä määrin muiden tekniikoiden kanssa. Se on vahvasti mukana työpöydän ohjelmien siirtämisessä verkkoon. AJAX kamppailee edelleen ongelmien kanssa, mutta niihin on löytynyt ratkaisuja. Tulevaisuus tuo mukaan uusia haasteita kilpailijoiden ja uusien virushyökkäysten muodossa. Niihin varaudutaan erilaisten yhteisöjen tietotaidon avulla ja pyritään kehittämään AJAXista turvallisempi ja entistä kilpailukykyisempi.

Tätä tutkimusta olisi mielenkiintoista tarkentaa koskemaan erityisesti Suomea. Miten suosittu AJAX on Suomessa ja mihin suuntaan sitä ollaan maassamme viemässä. Mielenkiintoista olisi seurata myös AJAXin maailmanlaajuista kehittymistä tulevaisuudessa. Sama tutkimus ensi vuonna olisi tuloksiltaan melko erilainen kuin tämä opinnäytetyö. Onko AJAX vielä numero yksi verkkosovellusten tuottajana vai kiiriikö Adobe takaisin ylimmälle pallille Apollon, Flashin ja Flexin voimalla? Entä miten reagoivat selaimet? Saadaanko AJAXille parempi tuki ja onnistutaanko estämään XSS-hyökkäykset?

LÄHTEET

Adobe Labs 2006. Apollo. [WWW-dokumentti].

<<http://labs.adobe.com/technologies/apollo/>>. (luettu 6.4.2007).

Ajax Patterns 2007. On-Demand JavaScript. [WWW-dokumentti].

<http://ajaxpatterns.org/On-Demand_Javascript>. (luettu 6.4.2007).

Aponovich, David, Lachlan, Donald, Langmaid, Nick, Magain, Matthew, Muir, Ian & Yank, Kevin 2006. State of Web Development 2006 / 2007. Australia: SitePoint Pty Ltd.

Backbase. [WWW-dokumentti]. <<http://www.backbase.com/demos/officelocator/>>.

(luettu 17.4.2007).

Basecamp. [WWW-dokumentti]. <<http://www.basecamphq.com/>>. (luettu 17.4.2007).

Bittiavaruus. HTTP-kutsu JavaScriptillä. [WWW-dokumentti].

<http://www.bittiavaruus.org/?page_id=20>. (luettu 6.4.2007).

Bosworth, Alex 2005. Ajax Mistakes. [WWW-dokumentti].

<http://sourcelabs.com/ajb/archives/2005/05/ajax_mistakes.html >. (luettu 10.2.2007).

Campbell, Bruce & Darnell, Rick 1997. Teach Yourself Dynamic DHTML in a Week. Indianapolis: Sams.net Publishing.

Cho, Kevin 2006. How to advertise on a pure AJAX application? [WWW-dokumentti].

< <http://www.jguru.com/forums/view.jsp?EID=1305379>>. (luettu 25.3.2007).

Crane, Dave – Pascarello, Eric – James, Darren 2006. Ajax in Action. Greenwich: Manning Publications Co.

- Eckel, Bruce 2007. How and Why AJAX, Not Java, Became the Favored Technology for RIAs. [WWW-dokumentti].
<<http://ajaxworldmagazine.com/read/333329.htm>>. (luettu 4.4.2007).
- Frye, Colleen 2006. Eric Pascarello dissects Ajax security vulnerabilities. [WWW-dokumentti]. <<http://searchwebservices.techtarget.com/qna/0,289202,sid26gci1164745,00.html>>. (luettu 25.3.2007).
- Google AdSense 2007. AdSense HelpCenter. [WWW-dokumentti].
<<https://www.google.com/adsense/support/bin/answer.py?answer=32725>>. (luettu 6.4.2007).
- Grossman, Jeremiah 2006. Myth-Busting AJAX (In)security. [WWW-dokumentti].
<http://www.whitehatsec.com/home/resources/articles/files/myth_busting_ajax_insecurity.html>. (luettu 25.3.2007).
- Heinonen, Oskari 2007. Document Markup. [WWW-dokumentti].
<http://www.cs.helsinki.fi/u/oheinone/teaching/xml_k07/pdf/slides-lect2.pdf>. (luettu 22.1.2007).
- Holzner, Steve 2006. Ajax For Dummies. Indianapolis: Wiley Publishing Inc.
- Kappe, Dietrich 2007. Ajax predictions for 2007. [WWW-dokumentti].
<http://blogs.pathf.com/agileajax/2007/01/ajax_prediction.html>. (luettu 4.4.2007).
- Kayak. [WWW-dokumentti]. <<http://www.kayak.com>>. (luettu 17.4.2007).
- Laszlo Systems Inc. 2006. OpenLaszlo the premier open-source platform for rich internet applications. [WWW-dokumentti]. <<http://www.openlaszlo.org/>>. (luettu 4.4.2007).
- Learningbusiness.fi 2006. OpenAjax Alliance vauhdittaa WEB 2.0 –teollisuutta. [WWW-dokumentti]. <<http://www.learningbusiness.fi/portal/31/?id=14132>>. (luettu 4.3.2007).

Lindroos, Jan-Erik & Lohivesi, Kari 2004. Onnistu Strategiassa. Helsinki: WSOY.

Linjama, Tero 2001. XHTML. Jyväskylä: Docendo Finland Oy.

Lipitsäinen, Arvo 2002. DOM – Document Object Model. [WWW-dokumentti].

<<http://myy.helia.fi/~atk90d/domARVO/dom.html>>. (luettu 13.2.2007).

Mahemoff, Michael 2006. Predictions: Ajax in 2007. [WWW-dokumentti].

<<http://ajaxian.com/by/topic/ajax/>>. (luettu 4.3.2007).

Netvibes.com. [WWW-dokumentti]. <<http://www.netvibes.com>>. (luettu 17.4.2007).

No Relation 2007. No future for AJAX? [WWW-dokumentti].

<<http://kyscorp.com/blog/2007/03/09/no-future-for-ajax/>>. (luettu 4.4.2007).

Nykänen, Ossi 2001. XML. Jyväskylä: Docendo Finland Oy.

Nykänen, Ossi 2003. XML-peruskurssi. [WWW-dokumentti].

<<http://www.w3c.tut.fi/training/2003/0219-xml-1/single.html>>. (luettu 22.1.2007).

Oasis. [WWW-dokumentti]. <<http://www.oasis-stores.com>>. (luettu 17.4.2007).

Pasztory, Adam. Flash vs. AJAX. [WWW-dokumentti].

<<http://www.pasz.com/articles/FlashVsAjax.html>>. (luettu 4.4.2007).

Peltomäki, Juha & Nykänen, Ossi 2006. Web-selainohjelmointi. Jyväskylä: Docendo Finland Oy.

Picard, Eric 2006. AJAX Counting Nightmares. [WWW-dokumentti].

<<http://www.clickz.com/showPage.html?page=3610786>>. (luettu 25.3.2007).

- Scott, Leland 2006. Ajax/DHTML Library Scorecard: How Cross Platform Are They?[WWW-dokumentti]. <<http://www.musingsfrommars.org/2006/03/ajax-dhtml-library-scorecard.html>>. (luettu 6.4.2007).
- Shah, Shreeraj 2006. Top 10 Web 2.0 Attack Vectors. [WWW-dokumentti]. <<http://www.net-security.org/article.php?id=949&p=2>>. (luettu 25.3.2007).
- Smarxwear. [WWW-dokumentti]. <<http://smarx.com/sample/smarxwear>>. (luettu 17.4.2007).
- Stenhouse, Mike 2005. Fixing the back button and enabling bookmarking for AJAX apps. [WWW-dokumentti]. <<http://www.contentwithstyle.co.uk/articles/38>>. (luettu 14.3.2007).
- Stewart, Ryan 2006. 10 Predictions for Rich Internet Applications in 2007. [WWW-dokumentti]. <<http://blogs.zdnet.com/Stewart/?p=204>>. (luettu 4.4.2007).
- Sunset Chat. [WWW-dokumentti]. <<http://blog.outer-court.com/chat/>>. (luettu 17.4.2007).
- Taft, Darryl 2007. Google's Bosworth: Why AJAX Failed (Then Succeeded). [WWW-dokumentti]. <<http://www.eweek.com/article2/0,1895,2088644,00.asp>>. (luettu 10.2.2007).
- The Mozilla Organization 2003. Introduction to the DOM. [WWW-dokumentti]. <http://web.archive.org/web/20040203025022/http://www.mozilla.org/docs/dom/domref/dom_intro.html>. (13.2.2007).
- The Mozilla Organization 2007. The Joy of XUL. [WWW-dokumentti]. <http://developer.mozilla.org/en/docs/The_Joy_of_XUL>. (luettu 6.4.2007).
- W3C 2007. Browser statistics. [WWW-dokumentti]. <http://www.w3schools.com/browsers/browsers_stats.asp>. (luettu 6.4.2007).
- Weewar.com. [WWW-dokumentti]. <<http://www.weewar.com>>. (luettu 17.4.2007).

Wikipedia 2007. Adobe Flex. [WWW-dokumentti].

<http://en.wikipedia.org/wiki/Adobe_Flex>. (luettu 4.4.2007).

Wikipedia 2007. Windows Presentation Foundation. [WWW-dokumentti].

<http://en.wikipedia.org/wiki/Windows_Presentation_Foundation>. (luettu 4.4.2007).

Zago LLC. [WWW-dokumentti]. <<http://www.zagodesign.com>>. (luettu 17.4.2007).

Zakas, Nicholas C – McPeak, Jeremy – Fawcett, Joe 2006. Professional Ajax.
Indianapolis: Wiley.

SANASTO

Alusta

Ohjelmisto- tai sovellusalusta (käytetään myös nimitystä kehitysalusta) on paikka, jossa julkaistaan ohjelmistoja. Käsitteellä viitataan usein käyttöjärjestelmään, mutta se voi toimia muullakin tavalla (esim. Java-alusta).

Appletti

Appletti on lyhyt Java-ohjelma, joka toimii WWW-selainympäristössä.

Asynkroninen pyyntö

Palvelimelle lähetetty pyyntö suoritetaan taustalla ilman sivun käytön keskeytymistä.

Attribuutti

Attribuutti on määrite (metatieto), joka täydentää elementtiä. Attribuutit merkitään elementin alkutagiin lainaus- tai heittomerkkien sisälle. Esimerkiksi `<body bgcolor="black"></body>`.

Avatar

Käyttäjän internetissä käyttämän hahmon visuaalinen esitys. Avataria käytetään mm. keskustelupalstoilla ja kolmiulotteisissa maailmoissa.

Bugi

Tietokoneohjelman lähdekoodissa oleva virhe.

DOM

DOM (Document Object Model) on ohjelmointikielestä riippumaton rajapinta, joka määrittelee HTML-elementit keskinäisen tiedon vaihdon ja sen miten elementteihin viitataan. Viittaamiseen DOM käyttää puurakennetta, joka lähtee juurisolmusta ja haarautuu useaksi alisolmuksi.

Dynaaminen sivusto

WWW-sivulla voi esiintyä visuaalista ja/tai sisällöllistä dynaamisuutta. Visuaalinen dynaamisuus on ruudulla näkyvää liikettä ja välkettä. Sisällöllisellä dynaamisuudella tarkoitetaan muuttuvaa sisältöä (esim. keskustelut ja uutisten päivittyminen).

Elementti

Elementit ovat dokumenttien rakentamiseen käytettyjä peruspalikoita, jotka voivat sisältää mm. tekstiä tai toisia elementtejä. Elementtien sijoittelu määrää dokumentin rakenteen ja sen avulla pystytään rajaamaan erilaiset sisällöt, kokonaisuudet tai rakenteet toisistaan.

Google Webtoolkit (GWT)

Googlen avointa koodia noudattava kehys, jossa AJAX-sovellukset kirjoitetaan Javalla. GWT kääntää Java-kielen JavaScriptiksi.

Hiekkalaatikko

Hiekkalaatikolla (sandbox) tarkoitetaan eristettyä ympäristöä, jossa ei-luotettavaa tai tutkimatonta ohjelmakoodia voidaan suorittaa turvallisesti rajoitetuilla käyttöoikeuksilla.

JSON

JavaScript Object Notation (JSON) on JavaScript-ohjelmissa käytetty datan vaihto - malli, jonka avulla asiakas- ja palvelinpuoli kommunikoivat keskenään. Kielen avulla pystytään lähettämään rakenteellista dataa eri ohjelmointikielillä tehtyjen ohjelmien välillä.

Jäsentäjä

Jäsentäjä (parser) rakentaa syötteenä saamalleen ohjelmalle jäsennykspuun. Jäsentäjä koostuu kokoelmasta funktioita, jotka tuottavat jäsennykspuun ylhäältä lähtien. Yksinkertaisimmillaan jäsentäjä toimii vain syntaksin tarkastajana.

Kehitysympäristö

Tunnetaan myös ohjelmointiympäristönä. Se on ohjelma, jolla ohjelmoija voi toteuttaa ohjelmiston. Yksinkertaisimmillaan ohjelmointiympäristö on tekstieditori ja ohjelmointikielen kääntäjän yhdistelmä.

Kehys

Sovelluskehys (käytetään myös ohjelmistokehys) muodostaa rungon sen päälle rakennettavalle tietokoneohjelmalle. Kehys toimii ohjelmoinnin apuvälineenä ja sen tarkoitus on nopeuttaa uusien sovellusten valmistusta. Kehys koostuu valmiiksi rakennetuista ohjelman osista, esimerkiksi tukiohjelmista, koodikirjastoista, skriptikielestä tai toisesta ohjelmistosta.

Kirjasto

Kokoelma valmiiksi tehtyjä toimintoja, jotka linkittyvät laadittavaan ohjelmaan.

Kääntäjä

Kääntäjän (compiler) avulla ohjelmointikielestä käännetään konekielinen ohjelma, joka suoritetaan. Kääntäjä tarvitsee avukseen jäsentäjän.

Merkintäkieli

Merkintäkielellä kuvataan dokumentin rakenne ja siihen kuuluvien tietoelementtien sisältöä, merkitystä ja ulkoasua. Osa merkintäkielistä on luonteeltaan metakieliä. Esimerkiksi HTML, XHTML ja XML ovat merkintäkieliä.

Metakieli

Metakieltä (käytetään myös kuvauskieli) voidaan käyttää tunnisteiden määrittelyyn ja niillä osoitettujen tietoyksikköjen sisällön luonnehdintaan. Sen avulla voidaan myös kuvata ja määritellä toinen kieli. Esimerkiksi XML on metakieliluonteinen merkintäkieli.

Ohjelmointirajapinta (API)

Käyttöliittymä, jonka avulla eri ohjelmat voivat tehdä pyyntöjä ja vaihtaa tietoja eli keskustella keskenään.

Olio-ohjelmointikieli

Ohjelmointikieli, jonka avulla tehdyt ohjelmat rakentuvat olioista. Oliot sisältävät toisiinsa loogisesti liittyvää tietoa ja toiminnallisuutta ja ne voivat kommunikoida keskenään lähettämällä ja vastaanottamalla viestejä.

Open Web Application Security Project

Avoin yhteisö, joka on omistautunut etsimään ja poistamaan ohjelmointiuhkia. Sen tarkoitus on tarjota yrityksille mahdollisuus kehittää, ostaa ja ylläpitää luotettavia sovelluksia. Kaikki projektin tarjoamat työkalut ovat ilmaisia ja avoimia.

Portaali

Verkkopalvelu, joka tarjoaa pääsyn useisiin muihin verkkopalveluihin. Portaali-sanaan on yleensä lisätty etuliitteeksi määritelmä portaalin luonteesta. Esimerkiksi uutisportaali tarjoaa pääsyn useille eri uutissivustoille.

Relaatiotietokanta

Relaatiotietokanta koostuu useista taulukoista, joiden tiedot esitetään riveinä ja sarakkeina. Tietokantaan tallennetaan myös tieto siitä, miten eri taulukot liittyvät toisiinsa.

RSS-syöte

RSS-muotoista syötettä luetaan RSS-lukijaohjelmilla. Ne vievät tietokoneelta vain vähän muistitilaa ja niiden avulla voi tilata esimerkiksi uutissivustojen uutissyötteen omalle koneelle. RSS-muodossa esitettävät tiedot ovat dynaamisia ja niitä käytetään mm. uutisten, tapahtumatietojen ja blogiviestien lähettämiseen.

Skriptikieli

Skriptikieli ts. komentosarjakieli on kieli, jolla toteutetaan tapahtumankäsittelyä ja muuta toiminnallisuutta HTML-dokumentteihin ilman ohjelman kääntämistä.

Skriptikieliä käytetään myös WWW-palvelinten ohjelmointiin.

Sovelluskehityskieli

Sovelluskehityskiehellä viitataan korkeatasoiseen perinteiseen ohjelmointikieleen, jolla kontrolloidaan tietokoneen toimintoja ja jonka avulla tehdään tietokoneohjelmia. Ohjelmointikielillä tehdyt ohjelmat on aina käännettävä ennen niiden suoritusta.

State of Development 2006/2007 -tutkimus

SitePointin ja Ektronin kesällä 2006 toteuttama tutkimus WWW-kehittäjien yhteisön tilasta. Tutkimukseen osallistui yli 5000 WWW-kehittäjää. Tutkimus selvitti WWW-kehittäjien työskentelytapaa, heidän käyttämiään tekniikoita ja tulevaisuuden suunnitelmia. Tutkimus julkaistiin lokakuussa 2006.

Tapahtumakäsittelijä

Ohjelman komponentit viestivät tapahtumakäsittelijälle minkälaisista muiden komponenttien lähettämistä viesteistä ne ovat kiinnostuneita. Tapahtumakäsittelijä käsittelee kaikki tuotetut viestit ja toimittaa ne niistä kiinnostuville komponenteille.

Tulkattavuus

Ohjelmakoodia ei ole käännetty suoritettavaksi tiedostoksi, vaan koodia tulkataan tulkin avulla ja suoritetaan sitä mukaan kun sitä tulkataan.

Tyypimäärittäminen

Tyypimäärittäminen tarjoaa keinon jakaa dokumentit hallittaviin aliluokkiin.

Tyypimäärittäminen asettaa elementtien ja attribuuttien nimet, arvojoukot ja rakenteet ja kuvaa säännöt, joita tietyntyyppisen dokumentin on noudatettava.

Validointi

Validoinnilla tarkoitetaan validaattorin suorittamaa tietojen oikeellisuuden tarkistamista. Tieto on validia, jos se on standardien mukaista.

Web 2.0

Nimitys, jota O'Reilly käyttää konferensseistaan, ja joka on levinnyt laajaan yleiseen käyttöön. Web 2.0 kuvaa nykyajan uusia ja kehittyneempiä web-sovelluksia. Web 2.0 mukaiset sivustot ovat mm. käytettävyydeltään erinomaisia, nopeita, yhteisöllisiä ja

muokattavia tietovarastoja. Esimerkiksi Flickr ja Google ovat Web 2.0 mukaisia palveluja.

Widget

Macintoshin käyttämä termi JavaScriptilla kirjoitetuille minisovelluksille (Microsoft käyttää termiä gadget). Opera-selaimessa termillä tarkoitetaan liitännäistä. Termillä kuvataan myös ohjelmistokomponentteja, joita käytetään verkkopalveluissa.

World Wide Web Consortium (W3C)

W3C on kansainvälinen yritysten ja yhteisöjen yhteenliittymä. Se ylläpitää ja kehittää verkossa käytettäviä standardinomaisia suosituksia. Jäseniä W3C:ssä on noin 360.

XHTML

XHTML on HTML-merkintäkielestä XML-kielen avulla laajennettua merkintäkieltä, joka noudattaa XML-kielen tiukkoja kielioppisääntöjä.

XML Namespaces

XML-nimijärjestelmä tai toiselta nimeltään XML-nimiavaruus on tekninen tunniste, jolla sovelluksessa käytettyjen termien merkitykset eri asiayhteyksissä voidaan luotettavasti erottaa.

XML Schema

XML Schemaa käytetään kuvaamaan XML-dokumentin rakenne. Se määrittelee dokumentin elementit, attribuutit ja niiden suhteen toisiinsa nähden.

XSRF

XSRF (Cross-site request Forgery) suorittaa hyökkäyksiä verkkosivustoihin. Hyökkäyksessä lisätään linkki tai koodin pätkä sivulle, joka viittaa esimerkiksi pankkipalveluun. Jos sivulle tulevalla toisella käyttäjällä on pankkiyhteys edelleen auki, yritetään vaarallista linkkiä ladata pankin palvelimelta ja samalla aiheutetaan esimerkiksi tilinsiirtotapahtuma.

XSS

XSS (Cross-site scripting) on hyökkäysmenetelmä, jolla pystytään suorittamaan käyttäjän selaimessa ohjelmakoodia hänen ollessaan yhteydessä luotettuun verkkosivustoon. Koska koodi on luotettavasta lähteestä, selain ei voi mitenkään erottaa hyökkäyskoodia muista koodeista. Hyökkäyskoodilla pyritään lähettämään hyökkääjälle käyttäjän tunnistamisessa käytettäviä evästeitä.

AJAXin kilpailijat

Kilpailija	Kuvaus	Miten kilpailee?	Vahvuudet / heikkoudet	Julkaisija
Flash	Kehitysympäristö	Tapa tehdä visuaalisia sovelluksia. Tällä hetkellä AJAXin pahin vastustaja.	<ul style="list-style-type: none"> - Hyvä animaatio- ja audio/videotuki - Vaatii Flash Playerin asennuksen - Huono ohjelmointiympäristö 	Adobe
Flex 2	Flashiin perustuva sovelluskehys	Flexillä on tuki grafiikoille ja taulukoille, joka AJAXilta puuttuu. Flexiä voidaan käyttää yhdessä AJAXin kanssa, joten se ei suoranaisesti ole sen kilpailija.	<ul style="list-style-type: none"> - Hyvä graafinen käyttöliittymä ja ohjelmointiympäristö - Ammattimaisen näköisiä sovelluksia, erityisesti bisneskäyttöön - Vaatii Flash Player 9:n asennuksen - Ilmaisversio ei ole niin kattava kuin maksullinen versio 	Adobe
Apollo	Ajoympäristö	Ei suoranaisesti kilpaile AJAXin kanssa, vaan pikemminkin Javan ja .NETin kanssa. Mahdollistaa työpöydän ohjelmien teon verkkotyökaluilla. Ohjelmat ovat myös verkkosovelluksia mitä kautta Apollo antaa AJAXille pienen haasteen.	<ul style="list-style-type: none"> - Toimii työpöydällä myös offline-tilassa - Yhdistää monta tekniikkaa (Flash, AJAX, HTML jne.) - Vaatii Flash Playerin asennuksen 	Adobe
XBAP	WPF-mallilla tehty selaimessa toimiva XAML-sovellus.	XBAB:t ovat rikkaita selaimessa toimivia sovelluksia. Kilpailukykyinen 3D-tukensa ansiosta. Häviää AJAXille huonon selaintukensa ansiosta.	<ul style="list-style-type: none"> - Hyvä 3D- ja grafiikkatuki - Vaatii .Net 3 kehityksen asentamisen - Vaatii Visual Studio 2005:sen - Vaatii muihin kuin Microsoftin selaimiin liitännäisen 	Microsoft
Atlas	Kehys	Haastaa AJAX-sovelluksissa käytetyn JavaScriptin C#-kielellä.	<ul style="list-style-type: none"> - Hyvä selaintuki - Turvallisuutta lisää se, ettei rajapintoja ole auki 	Microsoft
OpenLaszlo	Open Source -alusta	Samasta koodista saadaan sekä Flash-sovellus että AJAX-sovellus. Ei kilpaile suoranaisesti AJAX:n kanssa, vaan ennemminkin Flexin kanssa. OpenLaszlolla on AJAX-tuki.	<ul style="list-style-type: none"> - Avoin ja ilmainen - Taustalla toimiva ja tukea antava yhteisö - Toimii jo Flash 6 Playerissa. 	Laszlo Systems
XUL	Kuvauskieli	XUL:lla rakennetaan RIA-alustoja. Sillä on potentiaalia vaikka mihin, mutta huonon selaintukensa ansiosta se ei pärjää AJAXille.	<ul style="list-style-type: none"> - Nojaa standarditekniikoihin - Hyvä määrittely widgeteille - Toimii vain Mozillan selaimissa. 	Mozilla
Java-Applet	Minisovellus	Java-Appletteja käytetään lähinnä silloin, kun verkkosovellukseen tarvitaan ohjelmien kaltaisia ominaisuuksia. Ei ole kovin suosittu.	<ul style="list-style-type: none"> - Vaatii Java-liitännäisen - Raskas tekniikka - Hidas 	Sun Microsystems