

RIKKAAT INTERNETSOVELLUKSET

LAHDEN AMMATTIKORKEAKOULU
Mediatekniikan koulutusohjelma
Teknisen visualisoinnin suuntautumisvaihtoehto
Opinnäytetyö
4.5.2009
Miika Joronen

**Lahden ammattikorkeakoulu
Mediatekniikan koulutusohjelma**

JORONEN, MIKA: Rikkaat internetsovellukset

Teknisen visualisoinnin opinnäytetyö, 48 sivua, 1 liitesivu

Kevät 2009

TIIVISTELMÄ

Työn kirjallisessa osuudessa on lyhyesti kerrattu web-sovellukset sekä niiden tyyppisimmät ominaisuudet. Nopeasti kehittyvä tekniikka on tuonut internetsovelluksille paljon uusia mahdollisuuksia sekä vaatimuksia. Se on mahdollistanut muun muassa uuden tavan tarjota sisältöä monipuolisemmin internetin käyttäjille. Rikkaat internetsovellukset ovat tämän päivän nopeasti kehittyvä tekniikka, joiden periaatteisiin ja toimintaan perehdytään tässä työssä tarkemmin sekä esitellään niiden vaatimukset.

Adoben kehittämä käyttöjärjestelmäriippumaton AIR-ajoympäristö mahdollistaa työpöytäsovellusten toteuttamisen web-tekniikoilla. AIR-sovelluksia voidaan toteuttaa Adoben tarjoamalla työkaluilla sekä myös vaihtoehtoisilla työkaluilla. Työssä perehdytään Adoben tarjoamiin työkaluihin, mutta esitellään myös vaihtoehtoisia toteutustapoja.

Työn case-osassa toteutettiin kaksi erillistä RIA-sovellusta, jotka molemmat toimivat AIR-ajoympäristössä. Sovellukset luotiin lähtökohtaisesti perustuen Adoben tarjoamiin tekniikoihin ja ne kehitettiin tarkoituksella kahdessa eri kehitysympäristössä. Esimerkkisovellukset toteutettiin Flash- ja Flex-tekniikoilla.

Työni päätavoitteena oli perehtyä rikkaiden internetsovellusten toteutustekniikoihin ja vaatimuksiin sekä tutustua Adobe AIR-tekniikkaan ja sen tarjoamiin mahdollisuuksiin.

Avainsanat: Rikas internetsovellus, Adobe Flash, Adobe AIR, Adobe Flex

Lahti University of Applied Sciences
Faculty of Technology

JORONEN, MIKA: Rich Internet Applications

Bachelor's Thesis in Visualization Engineering, 48 pages, 1 appendix

Spring 2009

ABSTRACT

This Bachelor's thesis deals with rich Internet applications. The main goal was to become familiar with the execution techniques and requirements of rich Internet applications and to explore the Adobe AIR technique and possibilities provided by it.

The theoretical part of the thesis briefly presents Web applications and their most typical features. Rapid technical development has brought a lot of new possibilities and requirements for Internet applications. Advances in technology have enabled for example a new way to provide content more diversely for Internet users. Rich Internet applications are a fast developing technique of today, whose principles and functions are studied more in depth in the thesis. The requirements of the applications are also examined.

The non-operating-system-dependent AIR runtime environment developed by Adobe enables employment of web techniques in implementing desktop applications. AIR applications can be executed with tools provided by Adobe, as well as with alternative tools. The Adobe-provided tools are introduced and alternative methods of execution are also explored in the thesis.

In the empirical part of the thesis, example applications were made to demonstrate two powerful methods to develop rich Internet applications. The methods used were Flash and Flex mixed with AIR runtime environment.

Keywords: Rich Internet application, Adobe Flash, Adobe AIR, Adobe Flex

SISÄLLYS

TERMISTÖ

1	JOHDANTO	1
2	WEB-SOVELLUSTEKNIIKAT	2
2.1	Kehityshistoria	2
2.2	Yleiset sovellustekniikat	3
2.2.1	HTTP-protokolla	3
2.2.2	HTML-kieli	4
2.2.3	CSS-tyyliohje	4
2.2.4	JavaScript	5
2.2.5	DOM.....	5
2.2.6	ActiveX-kontrollit	5
2.2.7	Liitännäiset	6
2.2.8	Palvelintekniikat.....	6
2.3	Web-sovellukset.....	6
2.4	Toimintaperiaatteet	7
2.5	Web-sovellusten heikkoudet	7
2.6	Web-sovellusten edut	8
3	RIKKAAT INTERNETSOVELLUKSET.....	9
3.1	Yleistä	9
3.2	Historia.....	10
3.3	Käyttökohteet	10
3.4	Sovellusten rakenne	12
	<u>Käyttöliittymät</u>	13
3.5	Sovellustyypit.....	13
	<u>Selainpohjaiset web-sovellukset</u>	13
	<u>Liitännäisiin pohjautuvat web-sovellukset</u>	14
	<u>Scriptikieleen pohjautuvat web-sovellukset</u>	14
	<u>Web-toimitukseen pohjautuvat web-sovellukset</u>	15
3.6	RIA-sovellukselle asetetut vaatimukset	15
	<u>Suoritusympäristö</u>	19
	<u>Käytettävyys</u>	19

	<u>Sisällön haku</u>	20
	<u>Multimedia</u>	20
	<u>Kerrosittainen arkkitehtuuri</u>	20
	<u>CASE-välineiden tuki</u>	20
3.7	Vertailu web-sovellusten vaatimuksiin	21
4	ADOBE AIR	21
4.1	Yleistä	21
4.2	Adobe AIR:n toiminnallisuus	21
4.3	Edut ja heikkoudet	23
4.4	Tietoturva	24
4.5	Selain vai työpöytäsovellus	24
	<u>Sovelluksen asennus ja päivitys</u>	25
	<u>Käyttöjärjestelmätuki</u>	25
	<u>Toimintaympäristö</u>	25
	<u>Tiedon varastointi</u>	26
	<u>Verkon yhteydetön toiminta</u>	26
4.6	Kilpailevat tekniikat	27
4.6.1	Curl	27
4.6.2	Silverlight	28
4.6.3	JavaFX	29
4.6.4	OpenLaszlo	30
4.6.5	Mozilla Prism	30
5	SOVELLUSKEHITYSTYÖKALUT	32
5.1	Yleistä	32
5.2	Adobe Flash CS4	32
5.3	Adobe Flex Builder 3	33
5.4	Vaihtoehtoiset työkalut	34
5.5	Kehitystyökalun valitseminen	35
6	RIA-SOVELLUKSEN KEHITYSTYÖ	36
6.1	Toteutustekniikat	36
6.1.1	Flash-tekniikka	36
6.1.2	Flex-tekniikka	37
6.2	Käyttöliittymän suunnittelu	37

6.3	Multimedian käyttö sovelluksissa.....	37
7	CASE: RIA-SOVELLUS FLASH- JA FLEX-TEKNIIKALLA.....	38
7.1	Case sovellusten esittely	38
7.1.1	Flickr RIA	38
7.1.2	Web-selain RIA.....	40
7.2	Käytetyt tekniikat	41
7.3	Toiminnallisuus ja käytettävyys.....	42
7.4	Yhteenveto sovelluksista.....	42
8	YHTEENVETO.....	43
	LÄHTEET	45
	Painetut lähteet	45
	Sähköiset lähteet.....	45
	Kuvalliset.....	47
	LIITTEET	48

TERMISTÖ

AIR	Adobe Integrated Runtime
AJAX	Asynchronous JavaScript and XML
API	Application Programming Interface
COM	Component Object Model
CSS	Cascading Style Sheets
CASE	Computer-Aided Software Engineering
DHTML	Dynamic HTML
DOM	Document Object Model
ECMA	European Computer Manufacturers Association
HTML	Hypertext Markup Language
HTTP	HyperText Transfer Protocol
IDE	Integrated Development Environment
JavaEE	Java Enterprise Edition
JRE	Java Runtime Environment
JSP	Java Server Pages
PDF	Portable Document Format
PHP	Personal Home Page
RIA	Rich Internet Application
SDK	Software Development Kit
SOAP	Simple Object Access Protocol
SQL	Structured Query Language
TCP	Transmission Control Protocol
UI	User Interface
W3C	World Wide Web Consortium
WWW	World Wide Web
XML	Extensible Markup Language

1 JOHDANTO

Internetin kehitys on ollut nopeaa sen kaupallisesta julkaisusta 1990-luvulta lähtien. Nykyään se tarjoaa monipuolisen ja jatkuvasti kehittyvän pohjan monelle uudelle liikeidealle. Internet on tärkeä tiedonvälityskanava, joka on mullistanut maailmaa uudella tavalla tavoitettavuudella. Se on jopa niin iso osa monen ihmisen arkipäivää, että se on muuttanut ihmisten yleisiä toimintatapoja. Internetin sisältö on kokoajan kovan kehityspaineen alla ja sen staattinen esitystapa on alkanut muuttua yhä dynaamisemmaksi. Tekniikan kehittyessä ja vaatimusten kasvaessa on syntynyt paljon uusia tapoja toteuttaa sisältöä internetiin. Yksi näistä uuden sukupolven ratkaisuista ovat rikkaat internetsovellukset, jotka yhdistelevät uudella tavalla web- ja työpöytäsovellusten toimintatapoja rikkaan kokonaisuuden saavuttamiseksi.

Opinnäytetyössäni pyrin selvittämään, mitä rikkaille internetsovelluksilla tarkoitetaan ja mitkä ovat niiden käyttökohteet. Työssä pohditaan myös, mitkä ovat rikkaiden internetsovellusten vaatimukset ja mitä erilaisia toteutustekniikoita niiden kehittämiseen on tällä hetkellä tarjolla. Työssä tutustutaan aluksi web-sovelluksiin ja kerrataan tavallisimmat web-tekniikat. Rikkaiden internetsovellusten lisäksi työssä perehdytään myös rikkaiden internetsovellusten kehitystyökaluihin ja niihin liittyviin tekniikoihin.

2 WEB-SOVELLUSTEKNIIKAT

2.1 Kehityshistoria

Internetin historia ulottuu kauas 1960-luvun loppupuolelle, jolloin perustettiin internetin edeltäjä ARPANET. Network Core Protocol oli ARPANET:n ydin, joten internet oli vielä tuolloin nimenäkin tuntematon. Sähköpostin keksiminen vuonna 1972 käynnisti internetin kasvun, vaikka vielä tuolloin se tunnettiin ARPANET-nimellä. (Kaario 2002, 10.)

Internetin alkuperäinen käyttötarkoitus poikkeaa paljon siitä mihin se on tähän päivään mennessä kehittynyt. WWW-dokumentit tarkoitettiin alun perin tutkijoiden tiedonvälitykseen, jotta tutkimustietoa saataisiin välitettyä nopeasti ja tehokkaasti tekstimuotoisena käyttäjältä toiselle. Mahdollisuus luoda hyperlinkkejä WWW-dokumenttien välille mahdollisti myös sen, että dokumenteista pystyi liikkumaan helposti dokumentista toiseen. Tutkimustyöhön kehitettynä informaation jakelukanavana internet oli alkuperäisessä tarkoituksessaan todella onnistunut ja toimiva järjestelmä. Vuonna 1992 ilmestyi ensimmäinen graafinen selain Mosaic, minkä jälkeen internetin suosio levisi tiedeyhteisöjenkin ulkopuolle, ja pian 1990-luvun alkupuolella internet kaupallistettiin. (W3 2009.)



KUVA 1. Ensimmäinen graafinen selain Mosaic.

WWW:n kehitystä nopeuttivat sen alkuaikoina etenkin nopeasti kasvanut suosio, sekä uudet graafiset internetselaimet, kuten Internet Explorer ja Netscape Navigator. Sivustojen kehittyessä monipuolisemmiksi ja niiden tarjotessa tekstin lisäksi myös kuvia ja multimediaa, alettiin WWW:n mahdollisuuksia pohtia uudelta kantilta. Webtop oli tuolloin syntynyt visio siitä, kuinka internetiä voitaisiin hyödyntää suorittamalla niin sanottuja työpöytäsovelluksia suoraan palvelimelta web-selaimen välityksellä. Kehitys on kuitenkin tullut vasta nyt siihen pisteeseen, että tuollaisten webtop-sovellusten kehittäminen on mahdollista.

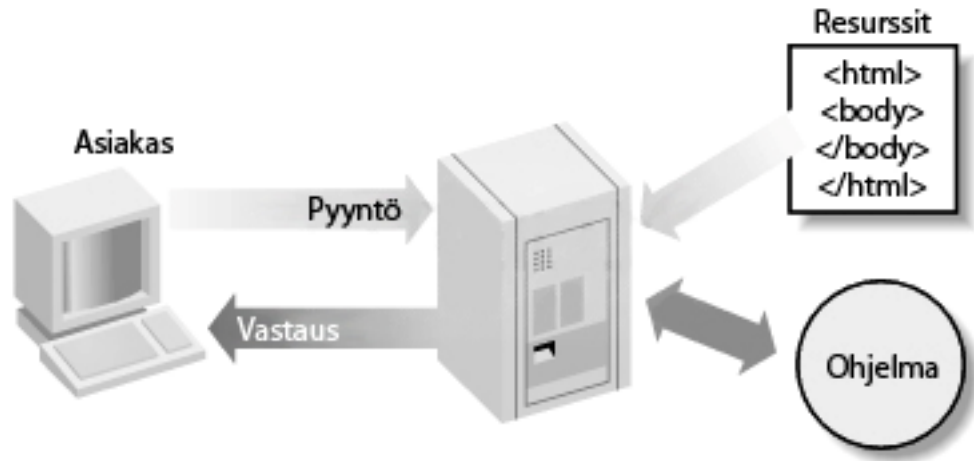
(Wikipedia 2009.)

2.2 Yleiset sovellustekniikat

Web-sovellusten toteutuksessa käytetään yleisesti useita eri toteutustekniikoita. Toteutustekniikat voidaan jakaa kahteen ryhmään, joita ovat palvelintekniikat sekä WWW-selaimessa näkyvään ulkoasuun ja käyttöliittymään liittyvät tekniikat.

2.2.1 HTTP-protokolla

HTTP on tilaton protokolla, jota selaimet sekä WWW-palvelimet käyttävät tiedonsiirtoon. Protokolla perustuu asiakasohjelman (esim. selain, hakurobotti) palvelimelle muodostamaan TCP-yhteyteen ja sen välittämään pyyntöön. Palvelin vastaa lähettämällä oikeanlaisen vastauksen, tavallisimmin esimerkiksi HTML-muotoisen sivun, binääridataa kuten kuvia, ohjelmia tai ääntä.



KUVA 2. Yksinkertainen HTTP-pyyntö.

2.2.2 HTML-kieli

HTML-kieli on WWW-dokumenttien julkaisua varten suunniteltu merkintäkieli. HTML:n avulla on mahdollista määrittellä dokumenttien rakennetta erilaisin elementein, hakea tietoa hypertekstilinkkien määrittelemistä kohteista, sekä suunnitella lomakkeita, joiden avulla on mahdollista kerätä syötteitä käyttäjältä sekä sisällyttää multimediaa dokumentteihin. Nykyisin on myös yleistynyt XHTML-merkintäkieli, joka perustuu XML-kieleen. XHTML-merkintäkieli mahdollistaa sivujen rakenteen, ulkoasun ja sisällön erottamisen paremmin toisistaan sekä helpottaa näin esimerkiksi sivujen päivitettävyyttä. (XHTML, W3 2009.)

2.2.3 CSS-tyyliohje

CSS-tyyliohjetiedostoja käytetään HTML- tai XHTML-dokumenttien ulkoasun määrittelyyn. CSS on monipuolinen merkintäkieli, jonka avulla internet selaimet osaavat muotoilla dokumentin ulkoasun käyttäjälle. Dokumentin sisältö ja rakenne voidaan helposti erottaa toisistaan, ja samaa tyyliohjetta voidaan käyttää myös muissa dokumenteissa. Tyyliohjeita käytettäessä ulkoasun ylläpito ja muokkaus helpottuvat, koska muutokset on mahdollista tehdä vain yhteen tiedostoon.

(Korpela 2008, 2-3.)

2.2.4 JavaScript

JavaScript mahdollistaa dynaamisten web-sovellusten toteuttamisen, ja se on myös hyvin yleiskäyttöinen ja kevyt ohjelmointikieli. JavaScript pohjautuu ECMAScriptikieleen, joka on JavaScriptin standardoitu versio. ECMAScriptikieli on suunniteltu sovellusriippumattomaksi scriptikieleksi, ja sen standardista vastaa European Computer Manufacturers Assosiation. Kieli on jaettu kahteen osaan: sovellusriippumattomaan runkokieleen ja sovelluskohtaiseen oliomalliin, jonka ansiosta sovellusriippumattomuus saavutetaan. (JavaScript, Wikipedia 2009.)

2.2.5 DOM

DOM on järjestelmäriippumaton ohjelmointirajapinta, joka mahdollistaa (X)HTML-dokumenttien sisällön muokkauksen. Liitettäessä JavaScript ohjelmointia sillä voidaan toteuttaa dynaamisia www-sivuja, jotka eivät kuitenkaan tarvitse jatkuvaa palvelinyhteyttä. DOM esittää dokumentin tiedot puumaisessa tietorakenteessa, jonka avulla niiden rakenteita on selkeä navigoida ja lisätä. Myös elementtien tai sisällön poistaminen ja muokkaaminen on selkeätä toteuttaa. W3C on määritellyt DOM-standardin. (Document Object Model, W3 2009.)

2.2.6 ActiveX-kontrollit

ActiveX on Microsoftin Windows-käyttöjärjestelmille kehitetty tekniikka, joka tunnetaan myös nimellä COM-tekniikka. COM-komponentit ovat uudelleenkäytettäviä ohjelmistokomponentteja. Microsoft nimesi markkinointisyistä vuonna 1996 osan Internet Explorer-selaimen yhteydessä käytettävästä OLE-tekniikasta ActiveX:ksi. OLE-tekniikka sai lopulta kokonaisuudessaan nimen ActiveX. ActiveX-kontrollit sekä liitännäiset voivat toimia vaihtoehtoisina tapoina toteuttaa samanlaisia laajennuksia selaimen. Selainlaajennukset voivat olla toteutettuna Microsoftin Internet Explorer selaimissa ActiveX-kontrolleina ja muissa selaimissa liitännäisinä. (Microsoft 2009.)

2.2.7 Liitännäiset

Liitännäiset ovat sovellusohjelmia, joiden avulla www-selain pystyy avaamaan tiedostot, joita se ei itse tue. Liitännäisiä on myös mahdollista liittää osaksi selainta. Tällä hetkellä yleisin liitännäisistä on Adoben Flash Player, joka mahdollistaa multimedian lisäämisen web-sivuille. Liitännäisiin perustuvat myös Java Appletit, mutta niiden merkitys ja käyttö ovat nykyään hyvin vähäistä. (W3 2009.)

2.2.8 Palvelintekniikat

Palvelintekniikat ovat selaimesta riippumattomia, koska ohjelman suoritus tapahtuu suoraan palvelimella. Palvelintekniikoilla toteutetaan web-sovelluksen palvelinosuus, ja käytetyimpiä tekniikoita ovat ASP.net, PHP, JavaEE ja Perl. Palvelinpuolella web-sovelluksilla on myös mahdollisuus käyttää tietovarastoinaan tietokantoja.

2.3 Web-sovellukset

Web-sovellukset ovat internetin välityksellä jaettavia ohjelmia, jotka toimivat www-selaimessa HTML-muotoisena. Ohjelmistotekniikan aloista web-sovellukset ovat nykyään yksi nopeimmin kasvavista aloista. Web-sovelluksen käyttöliittymä pohjautuu W3C:ssä määriteltyihin standardoituihin kuvaustekniikoihin. Web-sovellus toimii yhteistyössä web-palvelimen kanssa, toteuttaen sille määriteltyjä tehtäviä ja toimintoja. Kauppa-alalla ja liike-elämän palveluissa hyödynnetään yhä enenevässä määrin web-sovellusten tarjoamia mahdollisuuksia niiden vuorovaikutteisuuden ja tehokkaiden toimintatekniikoiden ansiosta. Web-sovelluksia käytetään nykyään myös usein osana yrityksen tietojärjestelmää, jolloin tietovarastoina käytetään web-sovelluksissa usein hyödynnettäviä tietokantoja tai XML-muotoista dataa. (Wikipedia 2009; W3 2009.)

2.4 Toimintaperiaatteet

Perinteiset web-sovellukset toimivat synkronisesti välittäen HTTP-muotoisilla pyynnöillä asiakkaan toiminnot palvelimelle, josta palvelin välittää vastauksen takaisin web-sovellukselle. Synkroninen toimintatapa tarkoittaa, että asiakas joutuu aina odottamaan palvelimen vastausta ennen kuin sovelluksen toimintaa voidaan jatkaa. Synkroninen toimintatapa voi pahimmillaan johtaa siihen, että asiakas joutuu keskeyttämään sovelluksen käytön, kunnes palvelin on toimittanut vastauksen takaisin sovellukselle. Viiveitä web-sovelluksen käytössä ei kuitenkaan normaalisti esiinny käytettäessä yksinkertaisia web-sovelluksia, mutta koska tänä päivänä kuitenkin vaaditaan yhä monipuolisempia toimintoja sekä ominaisuuksia, on tullut tarve ajatella web-sovellusten toimintatapoja uudella tavalla. Uudet tekniikat, jotka mahdollistavat Asynkroniset pyynnöt, ovat vastaus web-sovellusten toimintatapojen rikkaampaan ja tehokkaampaan toimitaan. (Building Responsive Web Applications 2006.)

2.5 Web-sovellusten heikkoudet

Heikkoutena web-sovelluksilla on tietoverkon vasteaika eli viive, joka aiheutuu palvelimelle lähetetystä pyynnöstä. Viiveeseen vaikuttaa kiinteästi internetyhteyden nopeus ja palvelimen kyky vastata web-sovellukselta tulleeseen pyyntöön. Työpöytäsovelluksia käytettäessä tämän kaltaista viivettä ei esiinny, vaan ne vastaavat käyttäjän toimintoihin välittömästi.

Eri selainten tavat tulkita web-sovelluksen toimintaa aiheuttavat myös nykyään vielä jonkin verran yhteensopimattomuusongelmia, vaikka useimmiten ongelmat voidaan kuitenkin välttää toteuttamalla web-sovellus tarkasti standardien mukaisesti. (Moock 2007, 15.)

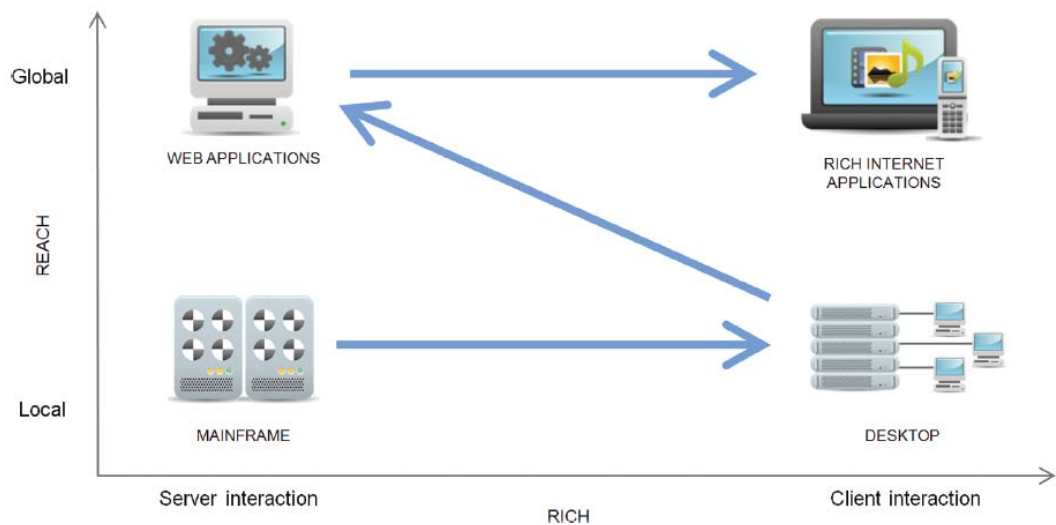
2.6 Web-sovellusten edut

Web-sovellusten eduksi voidaan lukea ensinnäkin niiden halpuus ja helppous ohjelmien jakelun suhteen. Web-sovellukset ovat kaikkien saatavilla, joilla on tietokone, ja internetyhteys. Yritysten ei esimerkiksi tarvitse asentaa ohjelmia erikseen jokaisen tietokoneeseen, vaan ne ovat kaikkien saatavilla internetin kautta. Web-sovellusten päivitys on myös työpöytäsovelluksiin verrattu huomattavasti kätevämpää, koska kun palvelimella oleva sovellus päivitetään uudempaan versioon, on se heti kaikkien käyttäjien saatavilla. Yhtenä suurena etuna on myös web-sovellusten käyttöjärjestelmän riippumattomuus, joka mahdollistaa sovelluksen käytön Mac-, Windows- tai Linux ympäristöissä, heti kun sovellus on asennettu palvelimelle. Jos sovellus on myös toteutettu oikeiden standardien mukaisesti, toimii se myös samalla lailla kaikissa nykyaikaisissa selaimissa, kuten Internet Explorer, Mozilla Firefox, Opera, Safari tai Google Chrome. Käytettäessä web-sovelluksia tietovarastot on myös mahdollista keskittää yhteen palvelimella olevaan tietokantaan useiden erillisten tietovarastoiden sijaan, mikä alentaa myös tietoturvariskiä. (Moock 2007, 16-17.)

3 RIKKAAT INTERNETSOVELLUKSET

3.1 Yleistä

Rikkaat internetsovellukset ovat web-sovelluksia, joiden ominaisuudet ja toiminnallisuus muistuttavat läheisesti perinteisiä tietokoneiden työpöytäsovelluksia. Rikkaissa internetsovelluksissa sovelluksen latauksen jälkeen suuri osa prosessoinnista tapahtuu selaimessa. Kommunikointi selaimen ja palvelimen välillä tapahtuu siis asynkronisesti. Rikkaiden internetsovellusten määrittelyyn kuuluu, että sovelluksen käyttöliittymä on toteutettu käyttämällä korkeatasoisia interaktio komponentteja. Sovellus voidaan toteuttaa niin, että se toimii joko web-selaimen päällä toimivilla tekniikoilla tai web-toimitusta hyödyntävillä tekniikoilla asiakkaan puoleiselta osaltaan. (Bozzon, Comai, Fraternali & Carughi 2006.)



KUVA 3. Rikkaiden internetsovellusten ja tavallisten web-sovellusten välinen saavutettavuus / rikkaus-kaavio.

3.2 Historia

Macromedia lanseerasi rikkaat internetsovellukset termin vuonna 2002 julkaisemassaan White paper-määrittelyssä. Määrittelyä oli kuitenkin käytetty jo vuosia aiemmin eri nimillä, mutta Macromedian lanseeraama Rich Internet Applications (RIA) -termi tuli viralliseksi määrittelyksi kyseiset vaatimukset täyttävälle web-sovelluksille. Edelleen käydään kuitenkin keskustelua siitä, mikä todella on rikas internetsovellus ja minkälaiset määrittelyt sen tulisi täyttää. (Macromedia 2002.)

3.3 Käyttökohteet

Rikkaat internetsovellukset ovat monimuotoisia ja tarjoavat perinteisten työpöytäsovellusten tehokkuuden sekä pyrkivät parantamaan sovelluksen käytettävyyttä. Käyttökohteina tämän kaltaisille sovellustyypeille ovat esimerkiksi verkkokaupat, online pörssi-sovellukset, multimediasovellukset ja monimuotoiset web-sovellukset yleisesti.

Yhdysvaltalainen huutokauppayhtiö eBay on esimerkiksi toteuttanut oman RIA-sovelluksen menestyksekkäästi tuoden verkkokauppansa myös käyttäjien työpöydille AIR-sovelluksen muodossa. eBay:n tapauksessa työpöytäsovelluksen avulla saavutetut edut ovat huomattavat esimerkiksi sovelluksen käyttöönoton ja käytettävyyden suhteen.



KUVA 4. eBay:n RIA-sovellus, joka on toteutettu Adoben AIR-tekniikalla.

Nasdaq oli myös ensimmäisten joukossa kehittämässä omaa RIA-sovellusta Adoben AIR-tekniikalla. Nasdaq kehitti Market Replay-nimisen RIA-sovelluksen käyttäen Adoben Flex-sovelluskehitystyökalua ja julkaisi sovelluksen AIR-muotoisena. Tämän RIA-sovelluksen avulla finanssialaan erikoistuneilla käyttäjillä oli mahdollista ensimmäistä kertaa seurata markkinoiden kehitystä tarkasti missä ja milloin vain. Sovelluksen tarjoamat tehokkaat työkalut markkinoiden seuraamiseen sekä graafisessa muodossa oleva informaatio, olivat tämän sovelluksen ansiosta kaikkien asiasta kiinnostuneiden tavoitettavissa.



KUVA 5. Nasdaq Market Replay Flex-sovellus AIR-ajoympäristölle.

3.4 Sovellusten rakenne

Kun rikas internetsovellus ladataan käyttäjän puolelle, sisältää se jo valmiiksi jonkin verran alustavaa tietoa. Latauksen jälkeen käyttäjä hallitsee itsenäisesti tiedon näyttämistä ja tapahtumien hallintaa. Sovellus kommunikoi palvelimen kanssa asynkronisesti vain, kun se tarvitsee lisää tietoa tai jos palvelimelle täytyy siirtää tietoa.

Rikkaiden internetsovellusten rakenteen toteutuksessa voidaan käyttää monenlaisia eri tekniikoita tai niiden yhdistelmiä. Adoben tarjoamia ratkaisuja ovat muun muassa Flash, Flex ja AIR-ajoympäristö. Vaihtoehtoinen toteutustekniikka on myös AJAX. Tässä työssä keskitytään kuitenkin tutkimaan rikkaiden internetsovellusten toteutusta lähinnä Adoben tarjoamilla työkaluilla ja tekniikoilla.

Käyttöliittymät

Rikkaat internetsovellukset mahdollistavat käyttäjälle monipuolisten käyttökokemuksen sekä sovellukset kykenevät suorittamaan vaativia tehtäviä ja toimintoja. Käyttöliittymän ulkoasu muistuttaa rikkaissa internetsovelluksissa toteutukseltaan ja toiminnaltaan työpöytäsovelluksia. Käyttöliittymän komponentit ovat korkeatasoisia ja monipuolisia ja niiden käyttö on yhtä tehokasta kuin työpöytäsovelluksissa. Sovelluksen rakenne muodostuu useimmissa tapauksissa palvelinsovelluksesta, tietokannasta sekä asiakassovelluksesta, joka ladataan palvelimelta ja suoritetaan selaimessa. Esityslogiikka ja osa liiketoimintalogiikasta on siirretty rikkaissa internetsovelluksissa asiakkaan puolella toteutettavaksi, joka vähentää merkittävästi asiakkaan ja palvelimen välistä tiedonsiirtoa. (Building Responsive Web Applications 2006.)

3.5 Sovellustyypit

Rikkaat internetsovellukset voidaan jakaa neljään ryhmään, jotka voidaan tunnistaa erilaisiksi sovellustyypeiksi. Sovellustyypejä ovat selainpohjaiset-, liitännäisiin pohjautuvat-, scriptikieleen pohjautuvat ja web-toimitukseen pohjautuvat web-sovellukset. Sovellustyypit ovat toteutustekniikaltaan suhteellisen erilaisia, ja niillä kaikilla on omat vahvuutensa.

Selainpohjaiset web-sovellukset

Selainpohjaiset web-sovellukset ovat usein toteutettu toimimaan jollain tietyllä web-selaimella. Tietylle selaimelle toteutettu web-sovellus kykenee tulkitsemaan ja renderöimään deklarativisesti määriteltyjä käyttöliittymiä.

Liitännäisiin pohjautuvat web-sovellukset

Liitännäisiin pohjautuvat web-sovellukset vaativat toimiakseen selaimen erikseen asennettavan liitännäisen, joka on usein kolmannen osapuolen tuottama. Liitännäisiin perustuvia web-sovelluksia ovat esimerkiksi Adoben Flash Player ja Microsoftin Silverlight, joiden avulla selaimet voivat esittää monimuotoisia web-sovelluksia.

Scriptikieleen pohjautuvat web-sovellukset

Rikkaat internetsovellukset, jotka perustuvat scriptikieleen on yleisesti toteutettu käyttäen JavaScriptiä. Sovelluksen käyttöliittymä toteutetaan HTML-kielellä ja CSS-tyyliohjelmäärityksillä, joita on mahdollista muokata scriptikielellä. Ajax on uudenlainen tapa yhdistellä olemassa olevia web-tekniikoita ja se perustuu Javascriptiin ja XML-kieleen. XMLHttpRequest-tuen yleistymisen myötä on yleisesti alettu puhumaan Ajax-sovelluksista. Ajax-termi ja konsepti esiteltiin ensimmäisen kerran vuonna 2005 Jesse James Garretin julkaisemassa artikkelissa: "Ajax: A new Approach to Web Applications". XMLHttpRequest-olio mahdollistaa palvelimen ja asiakkaan välisen kommunikaation asynkronisesti. Web-sovelluksen käyttöliittymä voidaan toteuttaa perustuen yhteen sivupohjaan, jossa käyttöliittymän eri osia päivitetään vain tarvittaessa, ilman että koko sivu tarvitsisi ladata uudelleen. Erillisiä selaimen asennettavia liitännäisiä ei tarvita käytettäessä Ajax-sovelluksia, sillä ne toimivat hyvin nykyaikaisilla internetselaimilla. Ajax-sovellusten käytettävyys ja saavutettavuus on hyvin korkea tasoista. (Building Responsive Web Applications 2006.)

Web-toimitukseen pohjautuvat web-sovellukset

Rikkaat internetsovellukset, jotka perustuvat web-toimitukseen ladataan internetin välityksellä käyttäjän koneelle, minkä jälkeen niitä suoritetaan selaimen ulkopuolella olevassa ympäristössä. Kun sovellus on ladattu käyttäjän koneelle, ei sitä tarvitse ladata enää uudestaan seuraavien suorituskertojen yhteydessä, ellei sovelluksesta ole julkaistu uutta versiota. Sovellukset käyttävät hyväkseen koneen verkkoresursseja ja mahdollisesti erillistä palvelimella olevaa sovellusta. Web-toimitukseen pohjautuvat sovellukset ovat hyvin samankaltaisia kuin työpöytäsovellukset, mutta sovellustekniikasta riippuen niiden pääsy paikallisiin resursseihin saattaa olla rajattua. Web-toimitukseen pohjautuvia tekniikoita ovat muun muassa Adobe AIR -sovellukset ja Java Web Start -sovellukset.

3.6 RIA-sovellukselle asetetut vaatimukset

Rikkaan internetsovelluksen vaatimukseen kuuluu, että sen tulee tarjota samanlaisia korkeatasoisia käyttöliittymäkomponentteja kuin normaalit työpöytäsovelluksetkin. Hyvän tuottavuuden omaava sovellus tarjoaa yhdenmukaisen käyttöliittymäkomponenttien käytön, jolloin sovelluksen käyttö on helppo omaksua. On toivottavaa, että rikkaan internetsovelluksen komponentit toimivat loogisesti verrattuna vastaaviin normaaleihin työpöytäsovelluksiin. Rikkaisiin internetsovelluksiin kehitetyt ajoympäristöt huolehtivat sovelluksen toistumisesta erilaisissa käyttöympäristöissä samalla tavoin, mutta myös muihin ratkaisuihin perustuvien sovellusten tulisi toimia samalla tavoin, vaikka niitä käytettäisiin erilaisissa päätelaitteissa.

Rikkaiden internetsovellusten tulisi tarjota käyttäjälle tehokas alusta sovelluksen suorittamien tehtävien toteuttamiseen. Työpöytäsovellusten interaktiivisen toteutuksen ja multimediaa hyödyntävien web-sovellusten yhdistämisestä muodostuu rikas kokonaisuus, jota perustellusti nimitetään rikkaaksi internetsovellukseksi. Rikkailta internetsovelluksilla viitataan usein myös hyvin toteutettuun interakti-

oon käyttäjän kanssa. (Bozzon, Comai, Fraternali & Carughi 2006; Macromedia 2002, White paper.)

Rikkaille internetsovelluksille voidaan tunnistaa neljä vaatimusta, mitä niiden tulisi täyttää:

- Ulkoasun yhteneväisyys: rikkaiden internetsovellusten tulisi käyttää samoja toimintaperiaatteita käyttöliittymän suhteen kuin työpöytäsovellustenkin, jotta käyttäjät omaksuisivat ne helpommin.
- Suoritusympäristön turvallisuus: käyttäjän puolella suoritettavan ajoympäristön päällä toimivan sovelluksen tulisi olla rakennettu niin, ettei se aiheuta käyttäjälleen tietoturvariskiä.
- Korkeatasoinen käytettävyys: käyttäjille tulisi tarjota korkeatasoisia käyttöliittymäkomponentteja, aivan kuten työpöytäsovelluksissa käytetyt interaktiiviset komponentitkin ovat.
- Lyhyt vasteaika: käyttäjien kannalta lyhyet vasteajat ovat tärkeitä, koska sovelluksen sujuva toiminta on siitä kiinni. Rikkaissa internetsovelluksissa tulisi käyttää Asynkronista tiedonsiirtoa, jotta kokonaisilta sivujen latauksilta vältyttäisiin.

Rikkaille internetsovelluksille sekä niiden toteutukseen liittyville tekniikoille voidaan esittää myös tarkemmin seuraavanlaisia vaatimuksia:

- Multimedia: mahdollisuus käyttää kuvaa, ääntä, videota, reaaliaikaista ja streaming multimediaa.
- Visuaalinen jatkuvuus: mahdollisuus välttää sovelluksen käyttöä häiritseviä ruudunpäivityksiä.
- Interaktio: mahdollisuus määrittellä, miten sovellus vastaa käyttäjän toimiin.
- Synkronisaatio: tuki tapahtumien määrittelyyn, jotka ovat joko ennalta määriteltäviä tai liittyvät käyttäjän toimiin.

- CASE-välineet: CASE-välineiden saatavuus, jotka helpottavat sovelluskehitystä.
- Kerroksittainen arkkitehtuuri: mahdollisuus jakaa sovellus erillisiin kerroksiin, joissa esimerkiksi tieto ja näkymät on eroteltu toisistaan.
- Dynaaminen sisällön lataus: mahdollisuus ladata sekä viedä sisältöä palvelimelle ajonaikaisesti.
- Rinnakkaiset pyynnöt useisiin tietolähteisiin: mahdollisuus ladata sisältöä yhtäaikaisesti käyttäen synkronista tai asynkronista-tiedonsiirtoa yhdestä tai useammasta lähteestä.
- Interaktiivinen kommunikointi: mahdollisuus reaaliaikaiseen kommunikointiin useiden eri käyttäjien kanssa.
- Personalisaatio: tuki universaalien sovellusten toteuttamiseen, jotka ovat hyvin saavutettavissa.

Rikkaiden internetsovellusten eroja muihin aiemmin esiteltyihin sovellustyyppeihin on vaikea havaita selkeiden standardien puuttumisen vuoksi. Kuva 6 esittelee rikkaiden internetsovellusten eroja muihin sovellustyyppeihin verrattuna yhdeksän ominaisuuden perusteella. (Bozzon, Comai, Fraternali & Carughi 2006; Macromedia 2002, White paper.)

Ominaisuudet	RIA-sovellus	Web-sovellus	Työpöytäsovellus
Sovelluksen asennus	Helppo	Helppo	Monimutkainen
Saavutettavuus	Hyvä	Hyvä	Huono
Interaktio ominaisuudet	Rikkaat	Rajoitetut	Kyllä
Palvelimen liiketoiminta-logiikka	Kyllä	Kyllä	Kyllä
Käyttäjän liiketoiminta-logiikka	Kyllä	Rajoitettu	Kyllä
Asynkroninen tietojen päivitys	Kyllä	Ei	Kyllä
Toistuva kommunikointi palvelimelle	Ei	Kyllä	Ei
Asiakas-palvelin kommunikaatio	Kyllä	Ei	Kyllä
Toiminta Offline-tilassa	Kyllä	Ei	Kyllä

KUVA 6. RIA-sovellukset verrattuna web-sovelluksiin ja työpöytäsovelluksiin.

Esitellyt rikkaiden internetsovellusten vaatimukset ovat osittain hieman sekavat ryhmittelynsä ja termistönsä vuoksi. Rikkaille internetsovelluksille on hyvin vaikea määritellä yksiselitteisiä vaatimuksia, sillä huolimatta useista lähteistä, niiden määrittelyä ei ole saatavilla kovinkaan kattavasti. Tällä hetkellä vaatimukset joita rikkaille internetsovelluksille on määritelty, pohjautuvat yleisesti eri tekniikoiden tarjoajien laatimiin selvityksiin. Kattavaa tieteellistä tutkimusta rikkaiden internetsovellusten vaatimuksista ei ole vielä julkaistu.

Sovelluksille asetetut vaatimukset ovat kuitenkin riittävän selvästi esitellyt, jotta niiden yhteneväisyyksiä voidaan tunnistaa. Seuraavaksi on jaoteltu rikkaiden internetsovellusten vaatimukset uudestaan niistä tunnistettavien pääkategorioiden

alle. Pääkategoriat ovat suoritusympäristö, käytettävyys, sisällön haku, multimedia, CASE-välineet ja kerroksittainen arkkitehtuuri.

Suoritusympäristö

- Suoritusympäristönä on esimerkiksi web-selain, joka on helposti käyttäjien saatavilla ja hyvin yleinen.
- Suoritusympäristöön asennettavan sovelluksen asennuksen tulee olla helppoa.
- Suoritusympäristön tulisi olla tietoturvan kannalta turvallinen ympäristö ja siellä suoritettavan sovelluksen tulisi suorittaa vain turvallista koodia.

Käytettävyys

- Käyttäjän toimiin on pystyttävä vastaamaan ja sovelluksen reagoititapaan on pystyttävä vaikuttamaan rikkaan interaktiotason saavuttamiseksi. Käyttäjän toimiin liittyvien tapahtumien sekä ennalta määrättyjen tapahtumien määrittämisen tulisi olla myös mahdollista.
- Sovellusten tulisi tarjota samanlaisia korkeatasoisia käyttöliittymäkomponentteja kuin työpöytäsovellustenkin. Sovellusten tulisi olla toimintalogiikaltaan ja käytänteiltään samantyylinen kuin työpöytäsovelluksen, jotta sen käyttäminen olisi mahdollisimman helppoa.
- Sovelluksen käyttöä häiritseviä ruudunpäivityksiä tulisi pystyä välttämään.
- Tuki universaalien sovellusten toteuttamiseen, jotka ovat hyvin saavutettavissa.

Sisällön haku

- Asynkronista tiedonsiirtoa tulisi käyttää sisällön hakuun, jotta vasteajat olisivat mahdollisimman lyhyet ja kokonaisten sivujen lataukselta vältyttäisiin.
- Ajonaikaisen tiedonsiirron tulee olla mahdollista asynkronisesti ja palvelimen kanssa tulee pystyä kommunikoimaan molempiin suuntiin.
- Tuki suorittaa samanaikaisesti useita rinnakkaisia pyyntöjä eri tietolähteisiin.
- Sovelluksen pitää pystyä toimimaan jollakin tasolla verkkoyhteyden katketessa tai jopa niin, ettei verkkoyhteyden katkeaminen vaikuta sovelluksen yleiseen toimintaan.

Multimedia

- Mahdollisuus käyttää kuvaa, ääntä, videota, reaaliaikaista ja streaming multimediaa.

Kerroksittainen arkkitehtuuri

- Mahdollisuus jakaa sovellus erillisiin kerroksiin, joissa esimerkiksi tieto ja näkymät on eroteltu toisistaan.
- Käyttäjän puolella tapahtuvaa web-toimitusta täytyy pystyä hyödyntämään tiedonsiirrossa.

CASE-välineiden tuki

- Käyttäjänpuoleisen toiminnan monimutkaistumisen vuoksi olisi tärkeää myös tukea käyttäjänpuoleista kehitystä CASE-välineiden avulla.

3.7 Vertailu web-sovellusten vaatimuksiin

Rikkaiden internetsovellusten vaatimukset ovat teknisesti huomattavasti vaativammat kuin web-sovelluksille esitetyt vaatimukset. Rikkaiden internetsovellusten vaatimusten perusteella voidaan perustellusti todeta, että termi rikas viittaa todellakin sovellusten ominaisuuksien ja sisällön monimuotoisuuteen.

4 ADOBE AIR

4.1 Yleistä

Adobe AIR on ajoympäristö, jonka käyttäjä pystyy luomaan rikkaita internetsovelluksia käyttäen tunnettuja web-sovellustyökaluja, kuten HTML, AJAX, Adobe Flash ja Adobe Flex. Adobe AIR on uuden mediatyyppin edustaja, jossa esiin tulevat web-sovelluksen sekä työpöytäsovellusten parhaat puolet. Web-suunnittelijat pystyvät Adobe AIR:n avulla luomaan täysin uudenlaisia sovelluksia, jotka toimivat kaikissa käyttöympäristöissä sekä online tai offline-tilassa.

Aiemmin työpöytäsovellusten ja web-sovellusten välillä ollut aukko on nyt saatu kurottua umpeen, sillä Adoben AIR tekniikka rakentaa puuttuvan sillan näiden kahden maailman välille. Adobe AIR:n pääelementit, kuten HTML-moottori, ActionScript Virtual Machine sekä paikallinen tietokantatoiminnallisuus SQLite perustuvat avoimeen lähdekoodiin. (Developing AIR Applications with Adobe Flash CS4 2009.)

4.2 Adobe AIR:n toiminnallisuus

AIR-ajoympäristö voidaan asentaa koneelle, joko ensimmäisen AIR-sovelluksen asennuksen yhteydessä tai erikseen. Tällä hetkellä uusimman käytössä oleva AIR:n versio on 1.5.1, joka julkaistiin helmikuussa 2009. Ajoympäristön koko on jonkin verran kasvanut päivitysten myötä, ja sen koko on tällä hetkellä noin 15MB, joka on kuitenkin edelleen huomattavan pieni verrattuna muihin ajoympäristöihin.

AIR:n toiminnallisuus muodostuu neljästä pääelementistä, jotka kaikki pohjautuvat avoimeen lähdekoodiin paitsi Adoben Flash Player. Elementit on valittu niiden tehokkuuden ja pienen koon vuoksi, minkä ansiosta koko ajoympäristö on saatu erittäin tehokkaaksi ja suhteellisen pienikokoiseksi.



KUVA 7. Adobe AIR-ajoympäristön pääelementit.

Webkit on HTML-moottori, joka vastaa HTML-, CSS- ja JavaScript-kielten tulkkauksesta AIR-sovelluksessa. Webkit perustuu avoimeen lähdekoodiin ja se on käytössä muun muassa Applen Safari selaimessa sekä Nokian S60-sarjan puhelinten internet selaimissa. Web-tekniikat jotka ovat tuetut Webkitin kautta, ovat XHTML, CSS, Javascript, W3C DOM Level 2 sekä XMLHttpRequest.

Flash Player on AIR-ajoympäristön elementti, joka vastaa Flash- ja Flex-tekniikoilla toteutetun sisällön esittämisestä. Flash-tekniikoilla toteutetut sisällöt ovat graafisesti kehittyneitä ja mahdollistavat muun muassa vektori-muotoisen grafiikan, videon, äänen ja 3D:n liittämisen sovellukseen.

Mozilla Foundationin avoimeen lähdekoodiin perustuva ECMAScript-tulkki Tamarin on AIR-ajoympäristön elementti, joka vastaa ActionScript tulkkauksesta. Tamarin tulee myös tulevaisuudessa vastaamaan Firefox-selaimen JavaScript-

tulkkauksesta ja tukee tällä hetkellä ECMAScript 3rd-versiota sekä osittain myös neljännen version määrittäjiä. (Mozilla Foundation 2009.)

SQLite mahdollistaa relaatiotietokanta toiminnon, joka on sisäänrakennettuna AIR-ajoympäristössä. Tietokanta tallennetaan suoraan käyttäjän koneelle eikä se vaadi erillistä palvelinta toimiakseen. SQLite pohjautuu C-kieleen, joten se on käännettävissä mille käyttöjärjestelmälle tahansa. SQLite mahdollistaa AIR-sovelluksille offline-tuen, jota on käytetty myös samaan tarkoitukseen esimerkiksi Google Gears:ssa tarjoamaan web-sovelluksille offline-tuen. (SQLite 2009.)

4.3 Edut ja heikkoudet

Adobe AIR mahdollistaa rikkaiden internetsovellusten toteuttamisen työpöytäsovelluksina käyttäen perinteisiä web-tekniikoita. Etuina AIR-sovelluksilla on tehokas ja kevyt ajoympäristö, mikä antaa laajan tuen käytettäville web-tekniikoille. AIR-ajoympäristö on käyttöjärjestelmäriippumaton, ja se on saatavilla Mac OS X-, Windows- ja Linux-käyttöjärjestelmille. Käyttöjärjestelmäriippumattomuuden ansiosta sovellukset toimivat samalla tavoin kaikissa käyttöjärjestelmissä, eikä sovelluksia tarvitse erikseen testata useilla eri selaimilla. Useissa vertailuissa Adoben AIR-tekniikkaa on pidetty tällä hetkellä kehittyneimpänä tapana tarjota rikkaita internetsovelluksia työpöytäsovellusmuotoisina.

Heikkouksina AIR-ajoympäristöllä ovat vielä sen heikko levinneisyys ja mobiililaitteiden tuen puuttuminen. Selainpohjaiset rikkaat internetsovellukset ovat paremmin käyttäjien saavutettavissa, sillä niiden käyttöön riittää pelkästään internet-selain. Tietoturva on myös yksi kriittisistä asioista mitä tulee huomioida käytettäessä AIR-sovelluksia, sillä ne omaavat muiden työpöytä-sovellusten tapaan kaikki samat käyttöoikeudet. Sovelluksen alkuperään tulee siis suhtautua aina hieman varauksella ja tutkia, voiko sen jakelijaan luottaa, ennen kuin aloittaa sovelluksen asennuksen.

4.4 Tietoturva

AIR:n tieturvamalli on kehitetty Flash Playerin tietoturvamallista lisäämällä siihen application-sandbox. Sandbox-termillä tarkoitetaan turvallisuusryhmiä, joihin eri tiedostot jaetaan. Tiedostoilla, jotka kuuluvat application-sandboxiin, on täydet oikeudet ajoympäristön tarjoamiin palveluihin ja rajapintoihin. Non-application-sandboxiin kuuluvat kaikki ne tiedostot, jotka ladataan sovelluksen ulkopuolelta ja ne omaavat rajoitetut oikeudet palveluihin sekä rajapintoihin.

AIR-sovellus suoritetaan samoilla oikeuksilla kuin tavalliset työpöytäsovelluksetkin. AIR-sovelluksen tietoturvatoteutus on siis erilainen, kuin tavallisten selaimessa toimivien web-sovellusten tietoturvamalli. Sovelluksen toimintaan vaikuttaa kuitenkin käyttöjärjestelmän käyttäjäkohtaiset rajoitukset. (Developing AIR Applications with Adobe Flash CS4 2009.)

4.5 Selain vai työpöytäsovellus

Internet on saavuttanut kasvavan aseman ohjelmistojen alustana tavallisten työpöytäsovellusten kanssa kilpaillen. Käyttöjärjestelmäriippumattomuus ja erinomainen saavutettavuus ovat selaimessa toimivien sovellusten suuria valtteja. Myös sovellusten helppo päivittävyys ja jakelu ovat lisänneet selainsovellusten suosiota. Kehittynyt tekniikka on kuitenkin mahdollistanut uudentyylisten työpöytäsovelluksia muistuttavien sovellusten toteuttamisen. Rikkaat internetsovellukset työpöydille tuova Adobe AIR-ajoympäristö yhdistää web-sovellusten ja työpöytäsovellusten parhaat puolet muodostaen uuden tehokkaan tavan toteuttaa sovelluksia.

Seuraavissa kappaleissa on vertailtu selainsovellusten ja AIR-työpöytäsovelluksen muutamia ominaisuuksia.

Sovelluksen asennus ja päivitys

AIR-sovellus on mahdollista ladata koneelle ja asentaa, kuten normaalit työpöytäsovelluksetkin, asennustiedoston kautta. Asennuksen voi kuitenkin myös suorittaa suoraan selaimen kautta. AIR-tekniikka mahdollistaa sen, että sovellus voi päivittää itsensä, jos sille on tarjolla uudempi versio. Selaimessa toimivat sovellukset mahdollistavat kuitenkin työpöytäsovelluksia helpomman tavan levittää sovellusta uusille käyttäjille. Asennusprosessia ei myös ole tarpeen suorittaa, koska sovellukset toimivat selaimen päällä. Selainpohjaisten sovellusten päivitys on myös työpöytäsovelluksia helpompaa, sillä päivityksen voi suorittaa vain palvelimelle, jonka jälkeen se on kaikkien käyttäjien saatavilla. (Adobe 2009.)

Käyttöjärjestelmätuki

AIR-ajoympäristö on tällä hetkellä saatavilla kaikille kolmelle yleisimmälle käyttöjärjestelmälle. Rikkaat internetsovellukset, jotka perustuvat liitännäisiin, vaativat erillisen liitännäisen asentamisen, jota ei välttämättä ole saatavilla kaikille käyttöjärjestelmille. WWW-selain, sen sijaan, löytyy tänä päivänä kaikista käyttöjärjestelmistä, minkä vuoksi web-sovellukset ovat yleisesti käyttöjärjestelmäriippumattomia. (Adobe 2009.)

Toimintaympäristö

AIR-sovellusten tietokoneen resurssien käyttöä ei ole rajoitettu. Sovellukset suoritetaan samoilla käyttöoikeuksilla kuin tavalliset työpöytäsovelluksetkin ja käyttöliittymä voidaan toteuttaa kokonaan halutunlaiseksi AIR-sovelluksesta käsin. Selaimessa toimivat sovellukset toimivat rajoitetussa toiminta ympäristössä eivätkä pysty hyödyntämään käyttäjän resursseja samalla tavoin kuin esimerkiksi AIR-sovellukset. Koska selainpohjaiset sovellukset toimivat selainsovelluksen päällä, on niiden käyttöliittymä aina riippuvainen selaimen sisältämistä toiminnoista ja käyttöliittymän muokkaaminen haluamansalaiseksi on mahdotonta. (Adobe 2009.)

Tiedon varastointi

AIR-sovelluksilla on kattavat mahdollisuudet muokata, tallentaa, luoda ja poistaa tiedostoja. AIR-ajoympäristö sisältää myös sisäänrakennetun tietokantatoiminnon, jota sovellusten on mahdollista myös käyttää tiedonvarastointiin. Selainsovelluksilla sen sijaan on hyvin rajoitetut mahdollisuudet tiedostojen tallennukseen käyttäjän koneelle. Selainsovellusten on kuitenkin mahdollista käyttää palvelinta tiedonvarastointiin.

Verkon yhteydetön toiminta

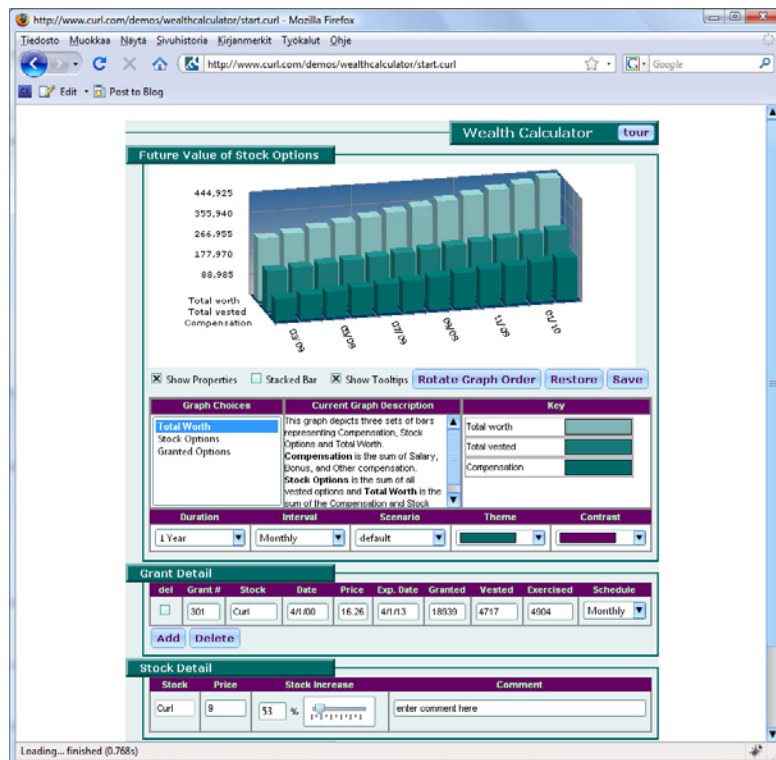
Työpöytäsovellukset hyödyntävät suurimmaksi osaksi koneen paikallisia resursseja, jonka vuoksi verkkoyhteyden katkeaminen ei pitäisi haitata sovelluksen toimintaa. AIR-sovellukset voivat tarkkailla verkkoyhteyden tilaa sekä luoda vapaasti uusia verkkoyhteyksiä. AIR-sovelluksilla on myös niin sanottu offline-tuki, johon sisäänrakennettu tietokanta sopii hyvin tietovarastoksi. Sovelluksen toiminta tulisi myös rakentaa niin, ettei se olisi täysin riippuvainen verkkoyhteydestä. Sen sijaan selainsovellukset ovat täysin riippuvaisia verkkoyhteydestä ja yhteyden katketessa senhetkiset tiedot yleensä menetetään. Offline-tuki on kuitenkin mahdollista saavuttaa käyttämällä selaimen asennettavaa liitännäistä, jonka avulla sivutiedot tallennetaan koneelle. (Adobe 2009.)

4.6 Kilpailevat tekniikat

Tällä hetkellä Adoben AIR-tekniikalla ei ole montaa kilpailijaa, jotka tarjoaisivat samanlaisia mahdollisuuksia rikkaiden internetsovellusten toteuttamiseen työpöydälle. Varteenotettavimmat kilpailijat AIR-tekniikalla ovat tällä hetkellä Curl, Silverlight, JavaFX, OpenLaszlo ja Mozilla Prism.

4.6.1 Curl

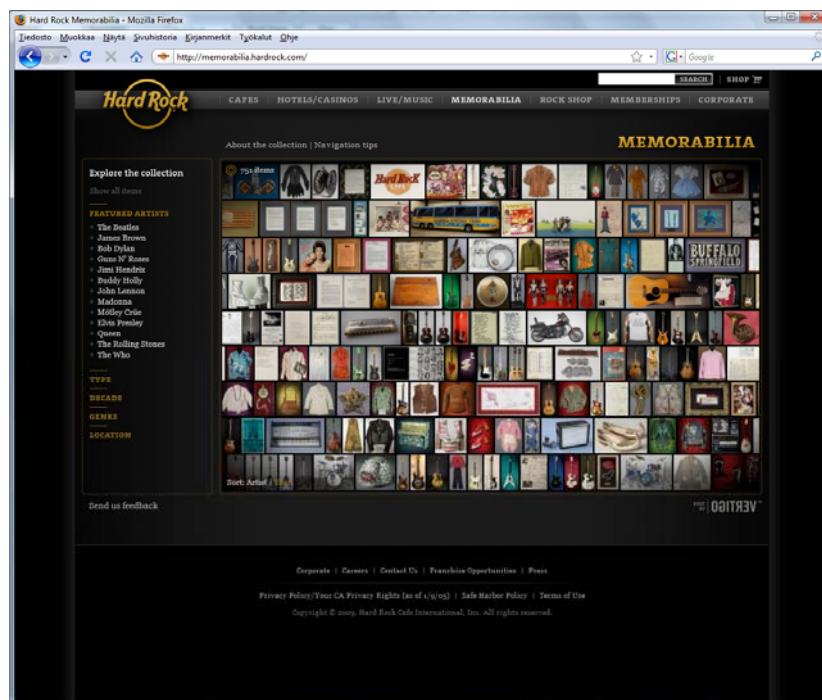
Curl on jo kauan alalla vaikuttanut ohjelmisto, joka tarjoaa valmiin tuotteen rikkaiden internetsovellusten toteuttamiseen liike-elämän tarkoituksiin. Curl ei tarjoa näyttäviä ja innovatiivisia ratkaisuja vaan pyrkii tarjoamaan helpon ja nopean tien rikkaiden internetsovellusten toteuttamiseen. Curl on hyvä työkalu yksinkertaisten ja informatiivisten sovellusten toteuttamiseen työpöytäympäristöön tai selaimen mutta ei kuitenkaan pysty kilpailemaan AIR:n tarjoamien uusien tekniikoiden kanssa.



KUVA 8. Rikas internetsovellus Curl-tekniikalla.

4.6.2 Silverlight

Silverlight on Microsoftin kehittämä selain- ja käyttöjärjestelmäriippumaton tekniikka nykyaikaisten selainsovellusten toteuttamiseen. Tekniikka mahdollistaa interaktiivisten web-sovellusten toteutuksen parempien käyttäjäkokemusten saavuttamiseen. Silverlight on tekniikkansapuolesta Flashin kovin kilpailija tällä hetkellä, mutta sen tarjoamat RIA-tekniikat ovat enemmänkin vanhempaa koulukuntaa verrattuna Adoben AIR-tekniikkaan. Microsoft esitteli MIX09 Web Design and Development -konferenssissa 18. Maaliskuuta, Silverlightin uusia ominaisuuksia, joihin tulee kuulumaan myös työpöytäsovellusten toteuttaminen. Silverlightin työpöytäsovelluksen toimintamalli on kuitenkin enemmänkin Google Gears-rajapinnan tyylinen ratkaisu. Silverlightiin pohjautuvat rikkaat internetsovellukset eivät siis sisällä yhtä laajoja mahdollisuuksia, kuin mitä Adoben tarjoamalla työkaluilla on mahdollista toteuttaa. Microsoftin RIA-tekniikka pohjautuu todennäköisesti jatkossakin selainympäristöön, jolloin sen käyttöympäristö on samaa tasoa kuin web-sovelluksilla. Silverlight-tekniikkaan pohjautuvilla rikkailla internetsovelluksilla on kuitenkin paljon potentiaalia, vaikka sen ratkaisut kilpailevatkin hieman eritasolla kuin AIR-tekniikkaan pohjautuvat sovellukset.



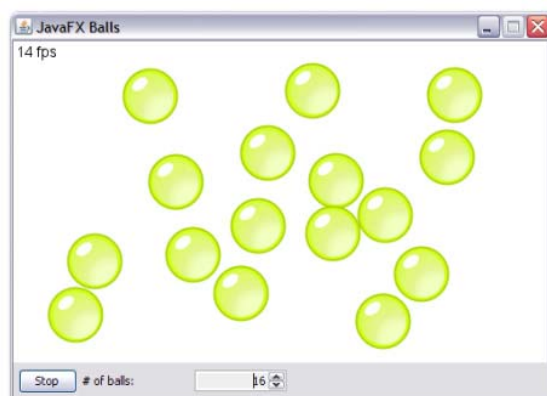
KUVA 9. Rikas internetsovellus Silverlight-tekniikalla.



KUVA 10. Silverlight työpöytäsovelluksen asennusruutu, jonka hyväksymällä sovellus on valmis käytettäväksi.

4.6.3 JavaFX

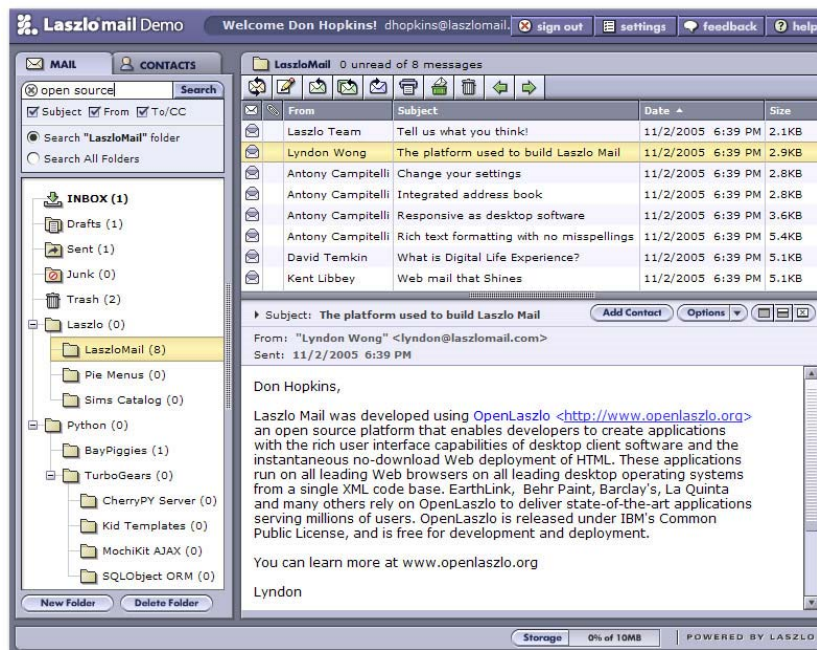
JavaFX on Sun Microsystemin kehittämä tekniikka rikkaiden internetsovellusten kehittämiseen. JavaFX on deklarativinen, staattisesti tyyppitetty skriptauskieli. Se käyttää Java2D Swing GUI -komponentteja, mutta on helpompi ohjelmoida kuin Java WebStart ja on myös sitä kevyempi. JavaFX:n toiminta perustuu ideaan, että sama sovellus pystyy toimimaan kaiken tyyppisissä päätelaitteissa. Sen rakenne koostuu JavaFX Scriptistä sekä JavaFX Mobilesta, jotka puolestaan ovat avoimen lähdekoodin alaisia tuotteita. JavaFX on vielä kuitenkin jäljessä rikkaiden internetsovellusten kehittämistekniikkana verrattuna esimerkiksi Flash- ja Silverlight-tekniikoihin, ja sitä on kritisoitu sen raskaan toteutuksensa vuoksi. JavaFX mahdollistaa kuitenkin rikkaiden internetsovellusten kehittämisen nykyaikaisin tekniikoin ja tuo ne myös käyttäjine työpöydille.



KUVA 11. Työpöytäsovellus JavaFX RIA-tekniikalla.

4.6.4 OpenLaszlo

OpenLaszlo on avoimen lähdekoodin ilmainen ohjelmointialusta rikkaiden internetsovellusten toteuttamiseen. Sovellukset toteutetaan XML-kielen perustuvalla LXZ-kielillä sekä JavaScriptillä. OpenLaszlo-sovellusten toteuttaminen on käyttäjäystävällistä, ja tekniikka on hyvin samankaltaista kuin perinteisten web-sovellusten toteuttaminen HTML- ja JavaScript-kielillä. LXZ-kielillä ohjelmoitu rikas internetsovellus voidaan kääntää SWF-muotoiseksi Flash-sovellukseksi tai Ajax-sovellukseksi.

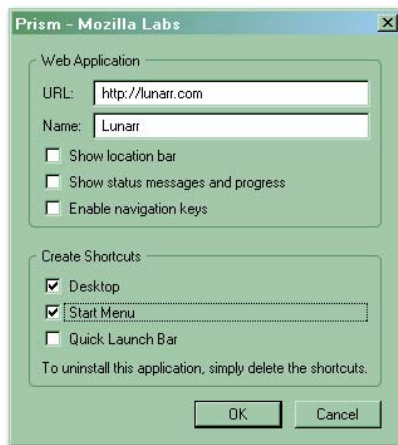


KUVA 12. *OpenLaszlo Mail, RIA-sovellus.*

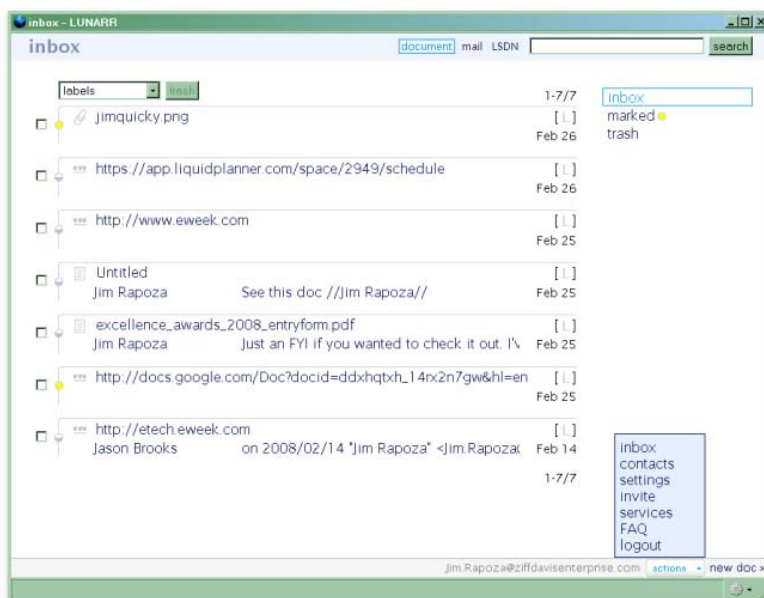
4.6.5 Mozilla Prism

Mozilla Prism on Mozilla Foundationin julkaisema tekniikka, jonka avulla web-sovellukset on mahdollista siirtää toimimaan selaimesta erillään olevassa ikkunas- sa ilman selainohjelmaa. Prism on käytännössä yksinkertaistettu versio selainik- kunasta, josta on poistettu kaikki selaimelle ominaiset käyttöliittymäkomponentit. Sovellus muistuttaa toiminnaltaan erehdyttävästi perinteistä työpöytäsovellusta,

mutta todellisuudessa sen toimintaympäristö ei poikkea selaimen toimintaympäristöstä. Mozilla Prismin toimintatapa poikkeaa paljon aikaisemmin esitellyn AIR-ajoympäristön toimintatavasta, koska sillä ei ole oikeuksia käyttäjän paikallisiin resursseihin vaan toimii selaimen rajoitetussa toimintaympäristössä. Mozilla Prism on yksinkertainen ja nopea tapa luoda rikkaita internetsovelluksia valmiiden web-sovellusten pohjalta.



KUVA 13. Mozilla Prism-sovelluksen asennus ikkuna.



KUVA 14. Työpöytäsovellus Mozillan Prism-tekniikalla.

5 SOVELLUSKEHITYSTYÖKALUT

5.1 Yleistä

Adoben RIA-teknologia muodostuu AIR-ajoympäristöstä, Flex- ja Flash-sovelluskehitystyökaluista sekä Flash Playeristä. Adoben RIA-teknologiat sisältävät työkaluja, kehitysympäristöjä, palvelimia, palveluja ja ajoympäristöjä, jotka toimivat hyvin yhteen. Näiden teknologioiden käyttöönottoa helpottavat myös useat Adobe Developer Connection opetusympäristöt, joissa kehittäjät voivat oppia nopeasti uusien teknologioiden käyttöä.

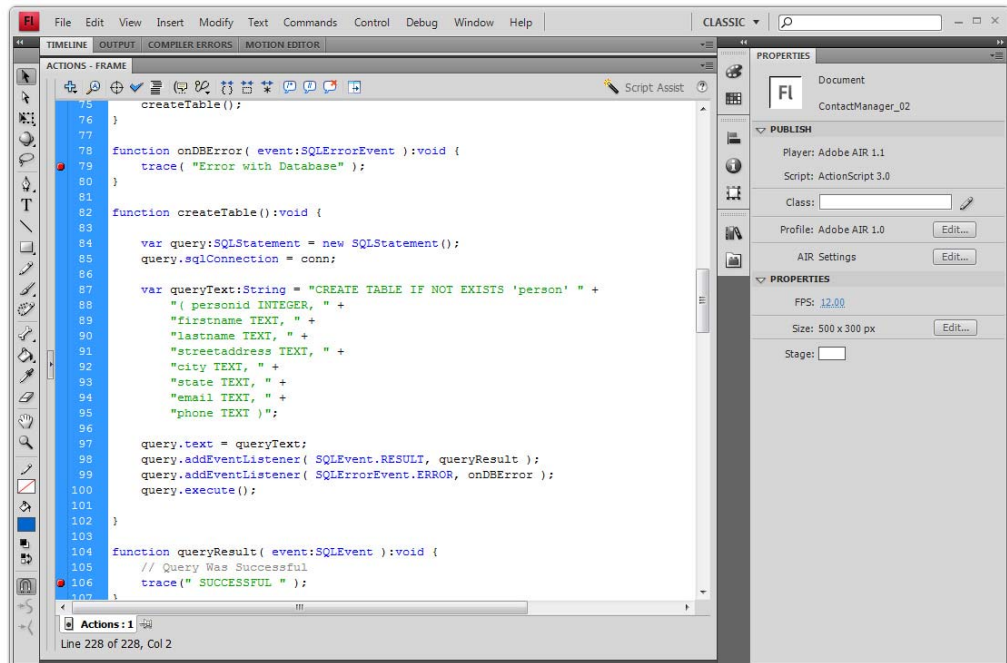
Rikkaiden internetsovellusten kehittämiseen on tällä hetkellä monia vaihtoehtoisia työkaluja, mutta markkinajohtajat Adobe ja Microsoft tarjoavat monipuolisimmat sekä kehittyneimmät kehitystyökalut. Tässä kappaleessa on selvitetty Adoben tarjoamien työkalujen käyttömahdollisuuksia kehitettäessä rikkaita internetsovelluksia. (Adobe 2009.)

5.2 Adobe Flash CS4

Adobe julkaisi syyskuussa 2008 uuden CS4 tuoteperheen, jonka myötä myös Flash CS4 julkaistiin. Flash CS4 toi mukanaan paljon uusia ominaisuuksia ja parannuksia edellisiin versioihin nähden. Rikkaiden internetsovellusten kehittäminen on tämän uuden ohjelmistopäivityksen myötä helpompaa kuin aikaisemmin, ja Adoben muut ohjelmat toimivat yhä paremmin yhteen sen kanssa. Tärkeimpinä uudistuksina rikkaiden internetsovellusten kehittämisen kannalta, ovat tuki AIR-sovellusten kehittämiseen, mikä puuttui aluksi kokonaan CS3-versiosta sekä parempi tuki Flex-ympäristön kanssa työskentelemiseen.

Adobe Flash CS4 soveltuu hyvin rikkaiden internetsovellusten kehittämiseen ja tarjoaa kehittäjälle monipuoliset työkalut näyttävien sovellusten aikaansaamiseksi. Flash soveltuu hyvin sellaisten sovellusten kehitykseen, joissa on runsaasti multim mediasisältö ja interaktiivisia komponentteja. Monet Flash-kehittäjät eivät

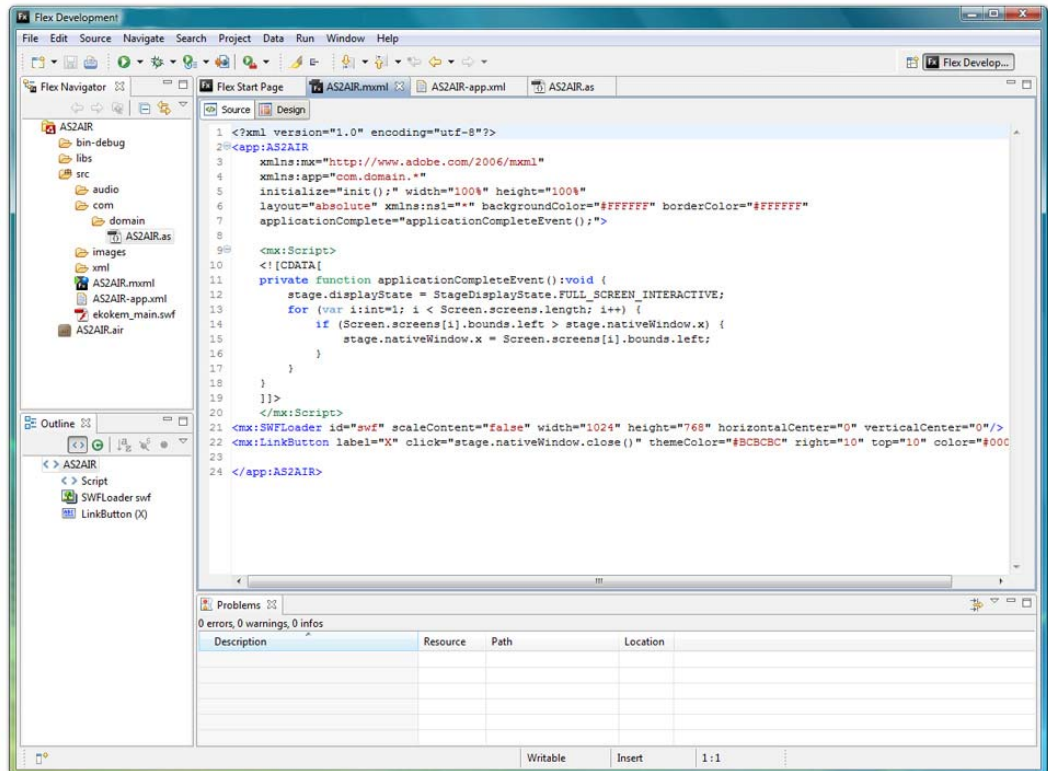
ole omaksuneet Flexin kehitystapaa, koska sen komponenttipohjainen työskentelytapa ei ole kaikissa tapauksissa paras ratkaisu. Flash ei sisällä samanlaista komponenttikirjastoa kuin Flex, minkä vuoksi sen käyttäjien täytyy tehdä monet Flex:ssä yksinkertaiset asiat kokonaan itse.



KUVA 15. Adobe Flash CS4:n käyttöliittymä.

5.3 Adobe Flex Builder 3

Adobe Flex 3 on kaikilla alustoilla toimiva avoimen lähdekoodin sovelluskehitys-ratkaisu rikkaiden internetsovellusten kehittämiseen. Adobe Flex Builder 3 on Eclipse-pohjainen kehitystyökalu, jonka avulla Flex-sovellusten kehitys on nopeaa ja suoraviivaista. Flex Builder 3 on kehitetty rikkaiden internetsovellusten kehittämiseen, ja sillä on kattava määrä ominaisuuksia RIA-kehitykseen AIR-tekniikan kanssa. Flex Builder 3:ssa on kattava määrä valmiita komponentteja, joita yhdistelemällä ja muokkaamalla kehittäjä voi nopeasti luoda näyttäviä ja korkean interaktiotason sisältäviä rikkaita internetsovelluksia.

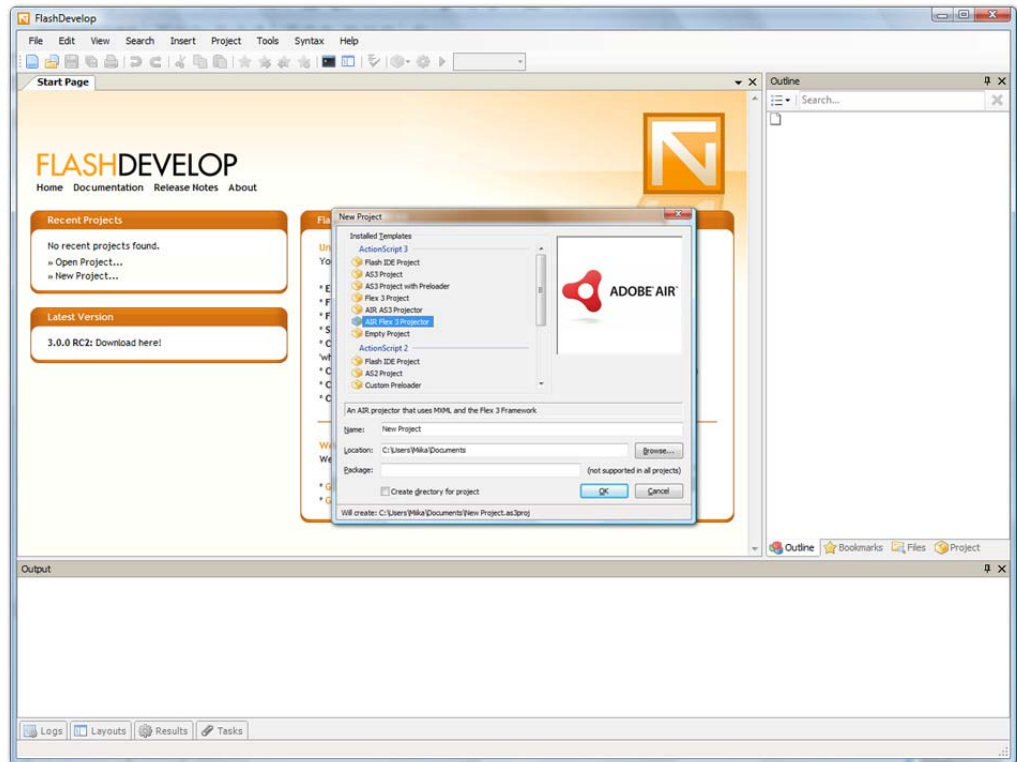


KUVA 16. Adobe Flex Builder 3:n käyttöliittymä.

5.4 Vaihteoiset työkalut

FlashDevelop on avoimeen lähdekoodiin perustuva ohjelma, jonka avulla on mahdollista toteuttaa Flash-, Flex- ja AIR-muotoisia projekteja. FlashDevelop on ulkoinen ActionScript editori, joka on käyttöominaisuuksiltaan jopa monipuolisempi kuin Adoben Flash ohjelmistoon sisällytetty editori. FlashDevelop tukee myös HTML, CSS ja XML tiedostomuotoja, mikä tekee siitä suhteellisen kattavan editorin. FlashDevelopiin on sisällytetty Flex 3 SDK, jonka avulla voidaan toteuttaa myös ActionScript 3 muotoisia Flex-projekteja.

Adobe on uuden Flash CS4 version myötä hieman parannellut editorin toimintoja, mutta FlashDevelopin monipuolinen tuki eri tiedostomuodoille tekee siitä erittäin varteenotettavan vaihtoehdon Flashin rinnalle.



KUVA 17. *FlashDevelop* käyttöliittymä.

5.5 Kehitystyökalun valitseminen

Oikean kehitystyökalun valitsemiseen vaikuttaa toteutettavan projektin vaatimukset ja minkälaisia elementtejä siihen tulee sisällyttää. Kehittäjä voi itse valita sopivimman työkalun rikkaan internetsovelluksen toteuttamiseen, ja koska niiden toteuttamiseen ei ole yhtä ja oikeaa ratkaisua, vaan toteutuksen voi rakentaa perustuen moneenkin eri tekniikkaan. Seuraavassa luvussa käsitellään tarkemmin mitä RIA-sovelluksen kehitystyössä tulee huomioida, kun valitaan sovelluksen toteutustekniikoita.

6 RIA-SOVELLUKSEN KEHITYSTYÖ

6.1 Toteutustekniikat

Rikkaan internetsovelluksen kehitykseen on tarjolla paljon hyviä työkaluja, joista osa on jopa ilmaisia. Sovelluksen toteutuksessa käytettävät tekniikat ratkaisevat hyvin pitkälti, mitä kehitystyökalua on hyvä käyttää. Jokaisella RIA-tekniikalla on omat vahvuutensa ja heikkoutensa, minkä vuoksi on hyvä perehtyä, mikä tekniikka soveltuisi työhön parhaiten.

Adobe tarjoaa Flash- ja Flex-tekniikoiden toteuttamiseen kehittyneet kehitystyökalut, joissa voi työskennellä joko suunnittelutilassa tai ohjelmointinäkyvässä.

Adoben tarjoamien tekniikoiden yhdisteleminen on järkevää ja joissain määrin jopa välttämätöntä parhaan mahdollisen lopputuloksen saavuttamiseksi. Tekniikat sopivat kuitenkin toteutustekniikoiksi hieman erityylyisiin RIA-sovelluksiin, joten on perusteltua tehdä tutkimusta sovelluksen vaatimuksista, ennen kuin kehitystyö aloitetaan jollain tietyllä tekniikalla.

6.1.1 Flash-tekniikka

Flash sopii hyvin toteutustekniikaksi sellaisiin RIA-sovelluksiin, jotka ovat sisältöään monimuotoisia ja sisältävät perinteisistä käyttöliittymäkomponenteista poikkeavia ratkaisuja. Flash sisältää Flex:n tavoin valmiita komponentteja, mutta useimmat Flash-kehittäjät käyttävät näitä valmiita komponentteja suhteellisen vähän. Monessa tapauksessa on helpompaa ohjelmoida jokin tietty komponentti kokonaan itse sen sijaan, että lähtisi muokkaamaan valmista komponenttia tarkoitukseen sopivaksi. Rikas internetsovellus, joka perustuu Flash-tekniikkaan, koostuu yleensä käyttöliittymästä, joka on toteutettu kokonaisuudessaan Flash-ympäristössä.

6.1.2 Flex-tekniikka

Flex Builder 3:ssa on kattava määrä valmiita komponentteja, joita yhdistelemällä ja muokkaamalla Flex-kehittäjä voi nopeasti luoda korkean interaktio tason sisältäviä rikkaita internetsovelluksia. Flex soveltuu hyvin tekniikaksi RIA-sovelluksiin ja pystyy käytännössä kaikkeen samaan mihin Flash-kehitysympäristössä toteutettavat sovelluksetkin. Flex on kuitenkin huomattavasti monipuolisempi kattavampi ympäristö toteuttaa rikkaita internetsovelluksia. Flex-tekniikka soveltuu hyvin teknisesti vaativiin sovelluksiin ja sellaisiin projekteihin, joissa sen kattavaa komponenttikirjastoa voi runsaasti hyödyntää.

6.2 Käyttöliittymän suunnittelu

Käyttöliittymän toteutuksessa on syytä muistaa rikkaille internetsovelluksille asetetut vaatimukset, vaikka ne eivät vielä tällä hetkellä standardin asemassa olekaan. Käyttöliittymän tulisi kuitenkin toteuttaa korkeatasoista käytettävyyttä ja tehokkuutta. Käytettäessä Flex-tekniikkaa on mahdollista hyödyntää komponentteja, jotka ovat yleisesti hyvin samankaltaisia kuin työpöytäsovelluksissakin ja saavuttavat näin myös korkean interaktiotason.

6.3 Multimedian käyttö sovelluksissa

Flash-tekniikalla on mahdollista toteuttaa monipuolisemmin sovelluksia, joissa on käytetty runsaasti multimediaa hyväksi. Flexin työkalut animaation ja multimedian toteutukseen ovat osittain rajalliset, ja tällaiset onkin parasta toteuttaa Flash-ympäristön puolella ja tuoda ne Flex-kehitysympäristöön. Yhteistyössä näillä kahdella tekniikalla saadaan aikaan todella näyttäviä ja teknisesti korkeatasoisia RIA-sovelluksia.

7 CASE: RIA-SOVELLUS FLASH- JA FLEX-TEKNIIKALLA

7.1 Case sovellusten esittely

Työn Case-osassa toteutettiin kaksi erillistä RIA-sovellusta, jotka molemmat toimivat AIR-ajoympäristössä. Lähtökohtaisesti sovellukset toteutettiin perustuen Adoben tarjoamiin tekniikoihin ja sovellukset toteutettiin tarkoituksella kahdessa eri toteutusympäristössä. Ensimmäinen esimerkkisovellus koostettiin Adobe Flex Builder 3 -ohjelmalla perustuen Actionscript 3-ohjelmointikieleen sekä MXML-merkkintäkieleen. Sovelluksessa hyödynnettiin myös AIR-ajoympäristön tarjoamia rajapintoja. Toinen esimerkkisovellus koostettiin Adobe Flash CS4 ohjelmalla perustuen Actionscript 3-ohjelmoitikielen ja AIR-ajoympäristön tarjoamiin rajapintoihin.

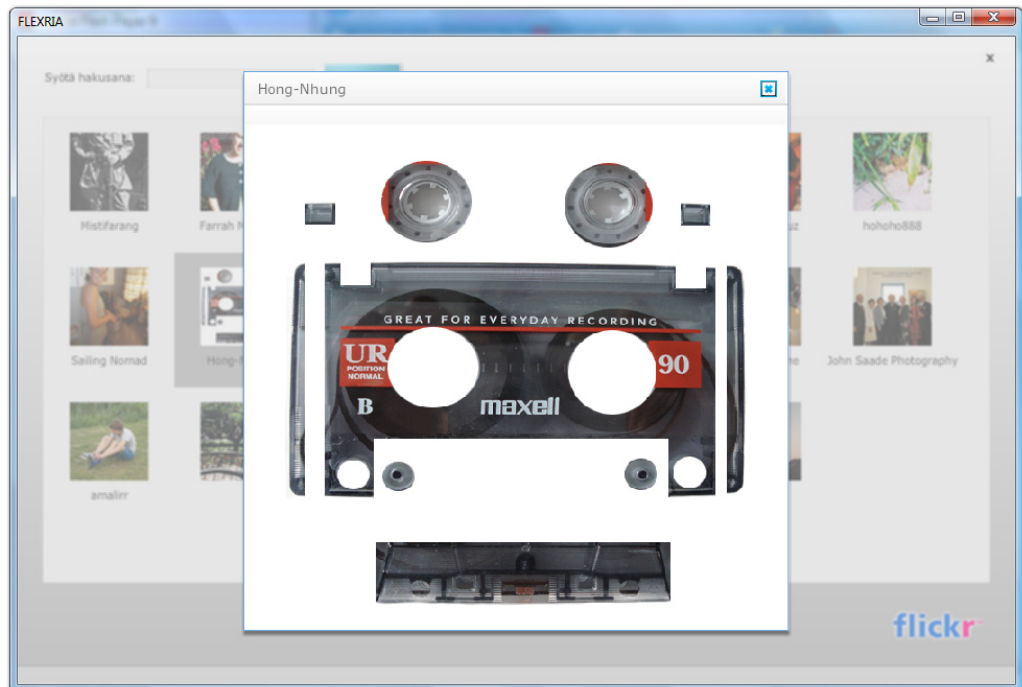
Sovellusten käyttötarkoitukset ovat suhteellisen erilaiset toisiinsa verrattuna, mutta sovellusten käyttöliittymien toteutuksessa pyrittiin samantyyliiseen lopputulokseen. Sovellusten toteutuksilla pyrittiin myös lähtökohtaisesti esittelemään tekniikoiden erilaisia soveltuvuuksia erityylisiin käyttötarkoituksiin.

7.1.1 Flickr RIA

Flickr on verkkopalvelu, jossa käyttäjät voivat julkaista ottamiaan valokuvia ja arvostella muiden kuvia. Flickrissä on mahdollista hakea ja luokitella kuvia folksonomisesti. Flickr on tällä hetkellä yksi internetin käytetyimmistä kuvapalveluista.

Flex-tekniikalla toteutettu sovellus on toiminnoiltaan kuvien katselu ohjelma, joka perustuu Flickr kuvien jakelusivustoon. Sovelluksella voidaan siis selata Etsi-toiminnon kautta Flickr-sivustolle ladattuja kuvia ja tallentaa niitä omalle koneelle.

Sovellus ottaa yhteyden verkkoon ja verkkoyhteyden ollessa saatavilla näkyy sovelluksessa sen merkiksi ilmoitus. Yhteydettömässä tilassa sovellus myös ilmoittaa verkkoyhteyden puuttumisesta. Sovelluksen käyttöliittymä koostuu kehyksestä, Etsi-kentästä ja Etsi-napista sekä Tallenna-napista. Etsi-nappia painamalla sovellus toteuttaa hakusana mukaisen kyselyn Flickr-sivustolle. Hakusanan mukainen sisältö ladataan Flickr-sivustolta ja se sijoitetaan kehyksen sisään sille varatulle alueelle. Kuvia voi selailla myös nuolinäppäimillä ja ne on mahdollista nähdä suurennettuna painamalla niitä hiiren vasemmalla painikkeella sekä myös tallentaa niitä käyttäjän koneelle painamalla Tallenna-painiketta.



KUVA 18. Flickr kuvasivustoon perustuva RIA Flex-tekniikalla.

Sovelluksen rakenne koostuu MXML-tiedostoista jotka yhdessä CSS-tiedoston kanssa määrittelevät sovelluksen ulkoasun ja käyttöliittymäkomponentit. Sovelluksen toiminnallisuuteen vaikuttavat määrittelyt on sijoitettu yhteen as-tiedostoon jossa on Actionscript 3-ohjelmointikielellä on toteutettu sovelluksen toiminnot.

7.1.2 Web-selain RIA

Flash-tekniikalla toteutettu sovellus on toiminnoiltaan mukautettu web-selain, johon on yhdistetty Flashin mahdollistamia erityiominaisuuksia. Selain toimii Webkit-moottorin voimin joka tarkoittaa, että se käyttää samaa HTML-moottoria kuin esimerkiksi Applen Safari selain.

Sovelluksen käyttöliittymä koostuu selaimen perus navigaatio elementeistä ja käyttäjävalikosta, joka on piilotettuna kunnes se tehdään hiirellä aktiiviseksi. Sovellus tukee skaalausta ja on rakennettu vektorigrafiikkaan pohjautuen.



KUVA 19. Flash ympäristössä toteutettu web-selain RIA.

Sovellus koostuu pelkästään Flash-ympäristössä laaditusta tiedostosta, joka on koostettuna swf-muotoinen tiedosto. Sovelluksen sisältämät elementit on sijoitettu Flash-tiedoston sisällä omille kerroksilleen, ja sen toiminnallisuus on määritelty omalla kerroksellaan, mihin kaikki toiminnallisuuteen vaikuttava koodi on sisälly-

tetty. Sovelluksen kehitystyö tapahtui hyvin saman tyyliä kuin normaalin Flash-sovelluksen kehitystyökin, mutta AIR-rajapinnan tarjoamien mahdollisuuksien ansiosta sen toiminnot ovat tavallista Flash-toteutusta huomattavasti kattavammat. Ohjelmoinnissa käytetyn Actionscript 3-ohjelmointikielen lisäksi sovelluksen toteutukseen tarvittiin myös tuntemusta AIR-ajoympäristön toiminnoista ja metodeista.

7.2 Käytetyt tekniikat

Sovellusten toteutukseen käytettiin Flash- ja Flex-työkaluja, jotka tarjoavat hieman erilaisen lähestymistavan sovellusten toteutuksen koostamiseen.

Flex-tekniikan tarjoamat komponentit ovat hyvin käyttäjäystävällisiä ja tarjosivat nopean tavan koostaa esimerkkisovelluksen käyttöliittymän toiminnallisuus. Sovelluksessa käytetyt valmiit komponentit muokattiin ulkoasultaan uudestaan ja niiden toiminnallisuus ohjelmoitiin Actionscript 3-ohjelmointikielillä. Flex mahdollisti Flashiin verrattuna huomattavasti tehokkaammat työkalut sisältöelementtien asemointiin, sillä kaikkia toimintoja ei tarvinnut ohjelmoida itse. Esimerkiksi nappien asemointiin pystyi määrittelemään suoraan Flex Buiderin -toimintojen avulla, jolloin ne sai helposti kellumaan tietyssä suhteessa sovelluksen reunoihin nähden.

Flash-tekniikalla toteutettu sovellus rakennettiin kokonaisuudessaan Flash sovelluskehittämissä ja toteutuksessa ei käytetty valmiita komponentteja, vaan käyttöliittymä rakennettiin kokonaan itse. Sovelluksen toiminnallisuus ohjelmoitiin Actionscript 3-ohjelmointikielillä ja AIR-ajoympäristön rajapintoja hyödynnettiin muun muassa verkkotoimintojen osalta, jonka avulla web-selaimen toteuttaminen Flash-ympäristössä oli mahdollista.

7.3 Toiminnallisuus ja käytettävyys

Toteutettujen sovellusten käytettävyys on korkea tasoista, ja ne täyttävät rikkaiden internetsovellusten vaatimukset monella tasolla. Sovellukset perustuvat selkeisiin käyttöliittymäkomponentteihin, joiden käyttö on esteetöntä. Tiedonsiirto on tehokasta ja käyttäjä ystävällistä, sillä tiedonsiirrossa on käytetty asynkronista tiedonsiirtoa, jotta toiminta olisi nopeaa.

Toteutetut RIA-sovellukset on julkaistu AIR-muotoisina tiedostoina, minkä ansiosta niiden käyttöönotto on helppoa. Sovellukset asennetaan saman tyylistä kuin normaalit työpöytäsovelluksetkin, mutta asennusvaiheet ovat vain yksinkertaisempia. Asennettu RIA-sovellus voidaan käynnistää työpöydälle luodun pikakuvakkeen kautta, ja se on nopeasti käyttövalmis ilman, että koko sovelluksen latautumista tarvitsisi odottaa.

7.4 Yhteenveto sovelluksista

Sovellukset ovat toiminnoiltaan suhteellisen yksinkertaisia, mutta RIA-tekniikoita hyödyntämällä niiden toiminnallisuus saatiin nostettua korkealle tasolle. Käyttäjien on huomattavasti helpompi sekä mielekkäämpi käyttää RIA-tekniikalla toteutettuja sovelluksia niiden tarjoamien uusien menetelmien vuoksi. Adoben tarjoamilla tekniikoilla oli mahdollista kehittää nopeasti ja tehokkaasti RIA-sovellus, jonka toiminnallisuus on samaa luokkaa kuin työpöytäsovelluksenkin.

8 YHTEENVETO

Rikkaat internetsovellukset on kehitetty web-sovelluksiin kohdistuneiden vaatimusten pohjalta, ja niiden tehtävänä on täyttää web-sovellusten ja työpöytäsovellusten välissä oleva aukko. Rikkaat internetsovellukset ovat myös vastaus web-sovellusten käyttöä haitanneisiin puutteisiin. Rikkaat internetsovellukset ovat saaneet osakseen paljon huomiota ja ennakko-odotuksia, mutta todellinen tekniikoiden välinen kehityskilpailu on vasta alkanut. Tekniikoita rikkaiden internetsovellusten toteuttamiseen on tarjolla monia ja toteutustyyliä poikkeavat jonkin verran toisistaan. Rikkaiden internetsovellusten saavuttamasta suosiosta huolimatta alan tieteellinen tutkimus on vielä puutteellista, ja kiinteiden standardien puuttuminen aiheuttaa sekavuutta määrittelyihin.

Tämän työn tarkoituksena oli pyrkiä selvittämään, mitä rikkailta internetsovelluksilla tarkoitetaan ja mitkä ovat niiden käyttökohteet. Työn tarkoituksena oli myös selvittää rikkaiden internetsovellusten vaatimukset ja mitä erilaisia toteutustekniikoita niiden kehittämiseen on tarjolla. Työssä perehdyttiin aluksi perinteisten web-sovellusten toteutustapoihin ja vaatimuksiin. Niiden pohjalta rikkaiden internetsovellusten vaatimukset oli myös mahdollista määrittellä ja todeta, mitkä ovat web-sovellusten todelliset edut sekä heikkoudet.

Rikkaiden internetsovellusten tutkimus aloitettiin perehtymällä niiden historiaan ja määrittelyyn, minkä jälkeen niiden käyttökohteet oli mahdollista ymmärtää sekä perehtyä niihin tarkemmin. Lähteiden perusteella kootuista vaatimuksista oli mahdollista johtaa uudelleen järjestellyt vaatimukset rikkaille internetsovelluksille, ja vaatimuksia oli mahdollista vertailla aiemmin esitettyjen web-sovellusten vaatimuksiin.

Rikkaista internetsovelluksista todettiin, että ne ovat hyvin saavutettavissa internetin välityksellä ja yhdistelevät uudella tavalla työpöytäsovelluksen ja web-sovelluksen hyvät puolet. Rikkaat internetsovellukset toimivat joko web-selaimessa tai erillisessä ajoympäristössä, joka mahdollistaa niille laajemman toimintaympäristön. Rikkaat internetsovellukset tarjoavat myös kehittyneitä käyttö-

liittymäkomponentteja, minkä ansiosta niiden käyttöönotto tulisi olla hyvin vaivatonta.

Rikkaiden internetsovellusten vaatimusten todettiin olevan huomattavasti laajemmat kuin aiemmin esiteltyjen web-sovellusten vaatimukset. Tämän vuoksi todettiin, että rikkaat internetsovellukset poikkeavat aiemmista web-sovelluksista vaatimusten perusteella.

RIA-tekniikat tarjoavat selviä etuja toteutettaessa web-sovelluksia riippumatta lähestymistavasta. Konkreettisimpia etuja RIA-tekniikkaa käytettäessä on esimerkiksi sovelluksen käytön tehokkuus, koska käytettäessä asynkronista tiedonsiirtoa kommunikointi asiakkaan ja palvelimen välillä on vähäistä.

Lähitulevaisuudessa tulee rikkaiden internetsovellusten ympärillä tapahtumaan todennäköisesti paljon, sillä vaikuttaisi siltä, että nämä uudentyyppiset sovellukset ovat vielä osittain murrosvaiheessa. Tällä hetkellä alan tutkimus on vielä suhteellisen puutteellista, ja todennäköisesti alan markkinajohtajat tulevat jossain määrin sanelemaan tulevat RIA-standardit. Tulevaisuudessa rikkaiden internetsovellusten ympärillä tulee myös riittämään keskustelua ja tulee olemaan mielenkiintoista nähdä, kuinka RIA-tekniikka tulee kehittymään ja kuinka se vastaa haasteisiin.

LÄHTEET

Painetut lähteet

Kaario, K., 2002, TCP/IP protokollat, Jyväskylä, Docendo, Finland.

Korpela, J., 2008, CSS Verkkosivujen muotoilussa, Jyväskylä, Docendo, Finland.

Moock, C., 2007, Essential ActionScript 3.0, California, O'Reilly Media, USA.

Nople, J., Anderson, T., 2008, Flex 3 Cookbook, California, O'Reilly Media, USA

Sähköiset lähteet

About SQLite. About SQLite 2009 [viitattu 2009]. Saatavilla:

<http://www.sqlite.org/about.html>

Adobe and Mozilla Foundation to Open Source Flash Player Scripting Engine.

Adobe [viitattu 2009]. Saatavilla:

<http://www.adobe.com/aboutadobe/pressroom/pressreleases/pdfs/200611/110706Mozilla.pdf>

Adobe Open Source. Adobe 2009 [viitattu 2009]. Saatavilla:

<http://opensource.adobe.com>

AJAX and PHP –Building Responsive Web Applications, Adobe PDF eBook

2006 [viitattu 2009]. Saatavilla: <http://www.packtpub.com/ajax-php/book>

Ajax: A New Approach to Web Applications. Adaptive path [viitattu 2009]. Saa-

tavilla: <http://www.adaptivepath.com/ideas/essays/archives/000385.php>

A Little History of the World Wide Web. W3 2009 [viitattu 2009]. Saatavilla:

<http://www.w3.org/History.html>

Browser vs. desktop. Adobe 2009 [viitattu 2009]. Saatavilla:

<http://www.adobe.com/products/air/comparison>

Conceptual modeling and code generation for rich internet application. Bozzon, Comai, Fraternali & Carughi 2006 [viitattu 2009]. Saatavilla:

<http://portal.acm.org/citation.cfm?id=1145649>

Document Object Model (DOM) Level 3 Core Specification, W3 2009 [viitattu 2009]. Saatavilla: <http://www.w3.org/TR/DOM-Level-3-Core>

Document Object Model (DOM). W3 2009 [viitattu 2009]. Saatavilla:

<http://www.w3.org/DOM>

Internet. Wikipedia [viitattu 2009]. Saatavilla: <http://fi.wikipedia.org/wiki/Internet>

JavaScript. Wikipedia [viitattu 2009]. Saatavilla:

<http://en.wikipedia.org/wiki/JavaScript>

White paper -Macromedia Flash MX -A next-generation rich client. Macromedia 2002 [viitattu 2009]. Saatavilla:

<http://download.macromedia.com/pub/flash/whitepapers/richclient.pdf>

Developing Adobe AIR 1.5 Applications with Adobe Flash CS4 Professional. Adobe 2009 [viitattu 2009]. Saatavilla:

http://help.adobe.com/en_US/AIR/1.5/devappsflash

XHTML™ 1.0 The Extensible HyperText Markup Language (Second Edition).

W3 2009 [viitattu 2009]. Saatavilla: <http://www.w3.org/TR/xhtml1>

Kuvalähteet

Kuvat 1-17. Miika Joronen, 2009.

LIITTEET

Liite: CD-rom levy, pdf-versio opinnäytetyöstä sekä Case-sovellukset.