

SAIMAAN AMMATTIKORKEAKOULU
Tekniikka Lappeenranta
Tietotekniikka
Viestintä

Pasi Valkeila

MAINOSTOIMISTON SISÄISEN TOIMINNAN- OHJAUSJÄRJESTELMÄN KEHITTÄMINEN

Opinnäytetyö 2009

TIIVISTELMÄ

Pasi Valkeila

Mainostoimiston sisäisen toiminnanohjausjärjestelmän kehittäminen 33 sivua

Saimaan ammattikorkeakoulu, Lappeenranta

Tekniikka, Tietotekniikka

Viestintä

Ohjaajat: Yrjö Utti, Saimaan ammattikorkeakoulu, Janne Mäenpää, Mediakolmio Oy

Opinnäytetyön aiheena on mainostoimiston verkkopohjaisen projekti- ja tehtävähallintajärjestelmän kehittäminen vastaamaan toimiston uusia kasvaneita tavoitteita. Projektin tilaajana on lappeenrantalainen mainostoimisto Mediakolmio Oy ja päättötyö toteutettiin heidän toimintamalliinsa räätälöitynä.

Opinnäytetyön tuloksena on projekti- ja tehtäväjärjestelmän käyttöönotetut lisäosat, jotka parantavat asiakkuuksien ja muiden yrityksen kontaktien hallintaa, helpottavat tehtäviin ja projekteihin liittyvää asiakaskommunikaatiota ja mahdollistavat Mediakolmion toiminnan analysoimista projektien, tehtävien ja asiakkuuksien hallinnan osalta.

Tämän lisäksi opinnäytetyössä esitellään Adobe Flex -ohjelmointikehys ja siinä käytettävät tekniikat verkossa toimivien sovellusten toteuttamiseksi.

Asiasanat: Toiminnanohjausjärjestelmä, Adobe Flex, Mainostoimisto, Rich Internet Application, Web 2.0

ABSTRACT

Pasi Valkeila

Improvement of internal project management system in advertising agency

33 pages

Saimaa University of Applied Sciences, Lappeenranta

Unit of Technology, Information Technology

Communications orientation

Instructors: Yrjö Utti, Saimaa University of Applied Sciences, Janne Mäenpää,

Mediakolmio Oy

The subject of this thesis is about enhancing project management software in an advertising agency. The customer's project management software is created using Adobe Flex framework and PHP programming language. Databases are driven by MySQL engine. The customer of this thesis is Mediakolmio Oy from Lappeenranta.

The results of this thesis are new software modules that are built to the core of project management software, which makes communication to the customer easier and brings new features to project handling in Mediakolmio.

The thesis also presents Adobe Flex framework and its development techniques in creating Rich Internet Applications.

Keywords: Project management software, PM, Adobe Flex, Advertising agency, Rich Internet Application, Web 2.0

SISÄLTÖ

1 JOHDANTO	9
2 MEDIAKOLMIO	10
2.1 Mekko II	10
2.2 Mediakolmion tietoverkko	11
3 MEKKO II -TOIMINNANOHJAUSJÄRJESTELMÄ	11
4 KÄYTETYT MENETELMÄT JA TYÖVÄLINEET	13
4.1 Web 2.0	14
4.2 Rich Internet Application (RIA)	14
4.3 Adobe Flex	15
4.4 ActionScript 3	17
4.5 MXML	18
4.6 Flex-ohjelmien käyttö	19
4.7 Adobe AIR	19
4.8 JavaScript ja Ajax	20
4.9 Tietokannan käsittely	21
4.10 HTTPService	22
4.11 XML	23
4.12 AMF	24
4.13 Flexin tulevat versiot	25
5 JÄRJESTELMÄN TOTEUTUKSEN VAIHEET	26
5.1 Esitutkimus	26
5.2 Määrittely	26
6 RATKAISU	27
6.1 Kontaktitietotarpeiden kehittäminen	27
6.2 Tehtävätrafiikka	29
7 YHTEENVETO	30
KUVAT	32
TAULUKOT	32
LÄHTEET	33

KÄSITTEET

ABC-tavukoodi	ActionScript bytecode (ABC) on Adoben kehittämä tavukoodi
ActionScript	ActionScript on macromedian kehittämä ECMAScript-tyyppinen olio-ohjelmointikieli
Adobe Integrated Runtime	Adoben kehittämä käyttöjärjestelmäriippumaton ympäristö, jonka avulla alunperin internetkäyttöön suunniteltuja ohjelmia voidaan ajaa työpöytäohjelmana
AIR	Katso Adobe Integrated Runtime
Ajax	Asynchronous JavaScript And XML (Ajax) on joukko tekniikoita, joiden yhteiskäytöllä WWW-sivustoa voidaan muuttaa dynaamisemmaksi
Alpha-versio	Ohjelmaversio, joka on tarkoitettu lähinnä sisäistä testausta varten. Alpha version jälkeen julkaistaan beta-versio, joka usein levitetään myös ulkopuolisten testaajien käyttöön
AMF	Action Message Format on binäärimuotoinen dataformaatti, joka on tarkoitettu tiedonsiirtoon
AMFPHP	Avoimella lisenssillä toteutettu yhdyskäytävä AMF-kehitystyön helpottamiseen
Apple	Tietokoneita, ohjelmistoja ja pienelektroniikkaa valmistava yritys
array	taulukko, muuttujatyyppejä ohjelmoinnissa
ASP.net	Microsoftin kehittämä ohjelmointitapa, jonka avulla kehitetään dynaamisia verkkopalveluita
AV	Lyhenne sanoista audiovisuaalinen
AVM1, AVM2	ActionScriptin käyttämiä virtuaalikoneita
breakpoint	Flex Builder -ohjelmassa käytettävä tarkistuspiste johon ohjelman kääntäminen pysähtyy
build	Ohjelman käännösprosessissa tapahtuva rakennusvaihe
Cascading Style Sheets	WWW-sivujen kehittämiseen suunniteltu tyyliohjejärjestelmä

client-server	Ilmaisu, jolla kuvataan palvelimen ja sitä käyttävän tahon välistä suhdetta
ComboBox	Flexin valintalaatikkokomponentti
cross-domain-policy.xml	XML-tiedosto, joka määrittelee käyttöoikeudet palvelimella oleville datapalveluille
CSS	Katso Cascading Style Sheets
databinding	Tiedon sitominen komponenttiin
DataGrid	Flexin ruudukkonäkymäkomponentti
Document Object Model	Ohjelmointirajapinta, jonka avulla muokataan HTML-sivujen näyttökomponenttien sisältöä ja asetuksia
Document Type Definition	Dokumentin tyyppimäärittely.
DOM	Katso Document Object Model
DTD	Katso Document Type Definition
ECMAScript	ECMAScript for XML (e4x) on ohjelmointikielen laajennus, jonka avulla XML-kuvauskielen käsittely helpottuu
Eclipse	Alunperin Java-ohjelmointiympäristöksi toteutettu, laajennettavissa oleva avoimen lähdekoodin ohjelmointityökalu
event	Ohjelmallinen tapahtuma
Extensible Markup Language	Yleiseen käyttöön tarkoitettu datarakenteiden kuvauskieli
External-paketti	Flash-kirjaston paketti, joka sisältää verkkoselaimen käsittelyyn vaadittuja luokkia
FAULT	Ohjelmallinen virhetapahtuma
Firefox	Verkkoselainohjelma
Flash	Macromedian kehittämä animaatioiden kehitykseen tarkoitettu ohjelmointiympäristö
Flash Player	Flash-tiedostojen katseluohjelma, jonka laajennuksen voi liittää verkkoselaimen yhteeseen

Flash Player Debugger	Ohjelmien vikojen etsintää helpottava Flash Player -versio
Flashvars	SWF-tiedoston käyttämä ohjelmaparametrien välitystapa
Flex	Macromedian Flashin kokemusten pohjalta kehittämä RIA-sovellusten kehitykseen tarkoitettu ohjelmointiympäristö
Flex Builder	Eclipse-pohjainen Flex-kehitysohjelma
FlexLibs	Google Code -sivuston jakama ilmainen Flex-komponenttikirjasto
GET	HTTP-palvelun parametrinvälityskeino
Google Docs	Googlen kehittämä tekstin ja taulukoiden käsittely ohjelma, joka toimii täysin internetissä
Gumbo	Työnimitys Flex 4-ohjelmaversiosta
HTML	Hypertext Markup Language
HTTP	Protokolla, jonka avulla välitetään internetliikennettä, palvelimen ja asiakaskoneen välillä
HTTPService	Flex-luokka, jonka avulla välitetään HTTP-protokollaa käyttäviä viestejä
Hypertext Preprocessor	WWW-ohjelmoinnissa yleisesti käytetty palvelinpuolen skriptiohjelmointikieli
Hypertext Markup Language	WWW-sivujen toteuttamisessa käytettävä kuvauskieli
IBrowserManager	Flex-luokka
ILOG Elixir	ILOG:n kehittämä kaupallinen Flex-komponenttikirjasto.
IP-osoite	Osoite, jonka avulla jokainen internetiin liitetty tietokone tunnistetaan
Java	Ohjelmointikieli, jolla voidaan tehdä sekä työpöytäohjelmia että WWW-ohjelmia
JavaScript	Asiakaspuolen skriptiohjelmointikieli, jota käytetään yleisesti HTML-sivujen yhteydessä.
labelField	MXML-komponenttien käyttämä parametri

Linux	Tietokoneen käyttöjärjestelmä
Macromedia	Nykyisin Adoben omistama yritys
Mediakolmio	lapeenrantalainen mainostoimisto
Mac OS X	UNIX-pohjainen käyttöjärjestelmä, jonka on kehittänyt Apple Inc.
Microsoft	Ohjelmistoja valmistava yritys
Model-View-Controller	Model-View-Controller on kolmiosainen ohjelmakehitysmalli
MySQL	MySQL on SQL-tietokannan (Structured Query Language) hallintajärjestelmä
MVC	Katso Model-ViewController
MXML	XML-pohjainen kuvauskieli
nimiavaruus	Määritelmä joka sisältää ohjelmointi, tai kuvauskieleen kuuluvat ominaisuudet
object	Muuttujatyyppi
Practical Extraction and Report Language	Monipuolinen palvelinpuolen skriptiohjelmointikieli
Perl	Katso Practical Extraction and Report Language
PHP	Katso Hypertext Preprocessor
POST	HTTP palvelun parametrinvälityskeino
RemoteObject-luokka	ActionScript3-luokka, jonka avulla on mahdollista käsitellä ohjelmaluokkia WWW-palvelimelta
RESULT	Tulostapahtuma
Rich Internet Application	Internetohjelma, joka käyttäytyy samankaltaisesti kuin työpöytäohjelma
RIA	Katso Rich Internet Application
run-time	Ajonaikainen
Safari	Applen valmistama verkkoselain

SDK	Katso Software Development Kit
Shockwave Flash	Shockwave Flash on tiedostotyyppi jota käytetään Flash ja Flex -ohjelmissa
skripti	Jollain ohjelmointikielellä toteutettu toiminnallisuus, jota ei tarvitse kääntää konekieliseksi ennen suoritusta
Simple Object Access Protocol	Oliopohjaiseen tiedonvälitykseen suunniteltu protokolla
Software Development Kit	kokoelma työkaluja ohjelmointikehitystä varten
SOAP	Katso Simple Object Access Protocol
Structured Query Language	Structured Query Language (SQL) on tietokannan käsittelykieli
SWF	Katso Shockwave Flash
Structured Query Language	Tietokannoissa käytettävä kyselykieli, jolla hallitaan tietokannan sisältöä
SVG	Scalable Vector Graphics (SVG) on XML-pohjainen kaksiulotteisten vektorigrafiikoiden kuvaamisen tarkoitettu kuvauskieli
tagi	Kuvauskielessä oleva ohjelmälähdekoodin osa, joka on merkataan kulmasuluilla
text	Muuttujatyyppi
virtuaalikone	Virtual Machine, VM verkkoselainten yhteydessä toimiva virtuaalikone, joka tarjoaa ohjelmalle toimintaympäristön ja hoitaa ohjelman ja järjestelmän välisen yhteyden
Web 2.0	WWW:n seuraava kehitysaskel
WebService-komponentti	Flex-ohjelmakomponentti
WWW	Lyhenne sanoista World Wide Web
WYSIWYG	Lyhenne sanoista What You See Is What You Get. Käytetään ohjelmakehityksessä kuvaamaan suunnittelunäkymää, jossa näyttökomponentteja voidaan sijoitella ruudulla ilman ohjelmakoodin kirjoittamista
XML	Katso Extensible Markup Language

XMLHttpRequest- DOM-luokka jota voidaan Ajax-ohjelmakehityksessä käyttää
luokka datan siirtoon ilman verkkosivun päivittämistä

1 JOHDANTO

Lappeenrantalainen mainostoimisto Mediakolmio Oy otti kesäkuun alussa vuonna 2008 käyttöönsä uuden toiminnanohjausjärjestelmän nimeltään Mekko II. Tämä järjestelmä kehitettiin Mediakolmion oman työvoiman toimesta aikaisemman Mekko-version kokemusten pohjalta. Mekko II toimii perustana koko Mediakolmion toiminnalle. Sen eniten käytetyt ominaisuudet liittyvät tehtävien ja projektien seuraamiseen tuotannossa ja työntekijöiden pitämiseen ajantasalla heidän töissään tapahtuvista muutoksista. Opinnäytetyön aiheena on kyseisen, vasta käyttöön otetun järjestelmän jatkokehittäminen.

Opinnäytetyön aiheesta aloitettiin keskustelu helmi-maaliskuussa 2008 ja tässä vaiheessa tehtiin alustava päätös siitä, että se liittyy uuteen käyttöön otettavaan, Mekko II -toiminnanohjausjärjestelmään. Opinnäytetyön toteutuksessa käytetään Adobe Flex -toimintaympäristöä. Suuri osa työn toteutuksesta on tämän uuden tekniikan omaksumista. Mediakolmiosta käytetään myöhemmin myös nimitystä yritys. Opinnäytetyön tarkoituksena on kehittää käytössä olevaa Mekkoa uusilla ominaisuuksilla, jotka mahdollistavat asiakkaiden liittymän omiin tehtävätietohinsa sekä parannetun kontaktitallennuskannan.

2 MEDIAKOLMIO

Mediakolmio Oy on vuonna 1996 perustettu mainostoimisto, jonka asiakas- kantaan kuuluu pääasiassa eteläkarjalaisia yrityksiä. Mediakolmio työllistää toistakymmentä työntekijää, jotka tekevät luovaa työtä muun muassa graafista suunnittelun, konseptisuunnittelun, sähköisen viestinnän, sekä AV-tuotannon (audiovisuaalinen) alueilla. Mediakolmion uusi toimisto sijaitsee vuonna 2008 käyttöön otetussa maakuntagalleriassa Lappeenrannan keskustassa.

Opinnäytetyö toteutettiin kesäkuun 2008 ja huhtikuun 2009 välisenä aikana Mediakolmion tiloissa.

2.1 Mekko II

Mediakolmion käyttämä Mekko II on yrityksen sisäverkossa toimiva, selain- pohjainen toiminnanohjausjärjestelmä. Sen toiminnot on toteutettu käyttäen Adobe Flex- ja PHP-tekniikoita (Hypertext Preprocessor) sekä sen käyttämät tiedot on tallennettu Mediakolmion sisäverkon tietokantapalvelimelle.

Mediakolmion verkkosivuilla osoitteessa <http://www.mediakolmio.fi> Mekkoa kuvataan seuraavasti:

Kuinka toimitetaan joka viikko satoja tuotteita oikeisiin lehtiin, kirjapainoihin, tulostus- ja postituspalveluihin? Toimitusjohtajamme mennessä naimisiin parikymppiset merkonomit passitettiin kotiin. Piti siis keksiä jotain uutta.

Sen kummemmin suurentelematta, puhtaasti mainostoimistokäyttöön toteutettu Mekko-projektinhallintajärjestelmä pitää paitsi huolta tuotannon luovuudesta, hallitsee se myös resurssit. Kiinteissä asiakassuhteissa meillä on mahdollisuus päästää uteliaimmat seuraamaan töiden ajankäyttöä ja edistymistä reaaliajassa.

Onhan meiltä tätä kysytty ostettavaksi ja muutama lisenssi on myytykin. Mutta kyllä Mekko aina isäntänsä tuntee.

2.2 Mediakolmion tietoverkko

Mekko II sijaitsee Mediakolmion sisäverkossa eikä sillä ole rajapintaa internetiin. Verkon laitteet ovat Applen valmistamia pöytätietokoneita sekä kannettavia tietokoneita.

3 MEKKO II -TOIMINNAHOHJAUSJÄRJESTELMÄ

Mediakolmio siirtyi Mekko II:n käyttöön 2.6.2008. Tähän asti käytössä oli ensimmäinen versio Mekko-järjestelmästä, joka todettiin riittämättömäksi kasvaneelle henkilöstölle. Mekko toimii toiminnanohjausjärjestelmänä Mediakolmion asiakasprojekteissa. Mekon toteutuksessa keväällä 2008 hyödynnettiin tietoja edellisen Mekko-version toiminnoista ja sen käytöstä opituista asioista. Uuden Mekon käyttöönotto suunniteltiin siten, että toimintojen ja ominaisuuksien kehitystyötä voitiin jatkaa keskeytyksettä. Tämä tarkoittaa sitä, että Mediakolmion tuotannossa, myynnissä ja projektihallinnassa sekä muussa hallinnossa työskentelevät ihmiset toimivat uuden järjestelmän lopputestaajina tuotannollisten tehtävien ohessa.

Järjestelmän ytimenä voidaan pitää tehtäviä, joita tehdään asiakkaille ja joiden tekemisestä vastaa Mediakolmion työntekijä. Tehtävä voidaan jakaa vaiheisiin päivämääräperustaisesti sekä tuntiperustaisesti. Tehtävät voivat olla yksittäisiä tai niistä voidaan koota projektikonaisuuksia. Työntekijä merkitsee tehtävälle tekemänsä tunnit sekä tuote- ja alihankinnat. Kun tiedetään tehtävän aikataulu ja tuntimäärät, voidaan laskea työntekijän kuormitukset kullekin ajanjaksolle. Näiden tietojen perusteella myös työtehtävien priorisointi helpottuu.

Työntekijät on jaettu neljään käyttäjäryhmään, joiden käyttöoikeudet on kuvattu taulukossa 1.

Taulukko 1 Mekon käyttäjryhmät

KÄYTTÄJÄRYHMÄ	SELITE
Peruskäyttäjä	Käyttäjä voi tarkastella omia tehtäviään ja projekteja, joihin ne liittyvät sekä lisätä niihin liittyviä työmerkintöjä sekä tuotemerkintöjä.
Hallintokäyttäjä	Käyttäjä voi luoda asiakkaita, sekä tarkastella kaikkia projekteja sekä tehtäviä.
Pääkäyttäjä	Käyttäjällä on samat oikeudet kuin hallintokäyttäjälläkin. Käyttäjä voi käsitellä muiden käyttäjien kirjautumitietoja ja lisätä uusia käyttäjiä.
Kehittäjä	kts. Pääkäyttäjä

Suurin osa käyttäjistä on peruskäyttäjiä. Hallintokäyttäjiä on järjestelmässä alle viisi ja pääkäyttäjiä vain yksi. Käyttäjien lisäksi järjestelmällä on kehittäjä, joka vastaa julkaistun järjestelmän ongelmatilanteiden selvittämisestä ja tarvittaessa pienten lisäosien luomisesta. Käyttäjiä ryhmitellään lisäksi asiakkaiden mukaan, jolloin heillä on oikeus luoda oman vastuusiakkaalleen projekteja ja tehtäviä.

Mekon käyttö tapahtuu verkkoselaimella. Suurin osa henkilöstöstä käyttää Safari-verkkoselainta, mutta myös Firefox-verkkoselaimia on käytössä. Mekon käytössä oleva versio on asennettu Mediakolmion sisäisen verkon palvelimelle, jossa ovat myös siihen liittyvät tietokantatoiminnot. Sisäverkkoon pääsy on estetty toimiston ulkopuolelta, jolloin tietoturvakysymyksien käsittely helpottuu.

Mekko II on toteutettu käyttäen Adobe Flex -ohjelmointikehystä sekä PHP- ja MySQL -ympäristöjä (Structured Query Language) tiedon tallennukseen käytetyissä toiminnoissa. Adobe Flex -ohjelmat käännetään SWF-tiedostoiksi (Shockwave Flash) rakennusvaiheessa, joten selaimilta, jotka sitä käyttävät vaaditaan vähintään Adobe Flash Player 9 -laajennus.

4 KÄYTETYT MENETELMÄT JA TYÖVÄLINEET

Perinteisessä internetohjelmoinnissa asiakaskoneen tehtävänä on ainoastaan muuttaa sille WWW-palvelimelta (World Wide Web) saatu data visuaaliseen muotoon. Tätä käytäntöä laajennetaan verkkoselaimessa esimerkiksi JavaScript-ohjelmointikielellä tehdyillä paikallisesti suoritettavilla skripteillä, joiden avulla tehdään yksinkertaisia tiedon tarkistuksia tai parannetaan WWW-sivun ulkoasua visuaalisilla tehosteilla. Tällaisen tiedonkäsittelyn tietoturva on tosin heikko, joten sen käyttö ja kohde on suunniteltava tarkoin.

Staattisista ja JavaScript-skripteillä parannelluista HTML-sivuista (Hypertext Markup Language) seuraavana askeleena pidetään erityyppisiä palvelinpuolen ohjelmakieliä, joista yleisiä ovat skriptiohjelmointikielet PHP ja Perl (Practical Extraction and Report Language). Nämä tekniikat käsittelevät datan WWW-palvelimella ja asiakaskoneen tehtäväksi jää edelleen ainoastaan tulostaa palvelimelta saatu edelleen käsitelty data katsojalle ymmärrettävään muotoon.

Uutena tiedon käsittelytapana verrattuna aikaisempiin on käytäntö, jossa asiakaskoneelle siirretään ensin raakadatan lisäksi sitä käsittelevät toiminnot ja tämän jälkeen jotkut tiedonkäsittelytoiminnot voidaan siirtää palvelimelta asiakaskoneelle. Tällaiset ominaisuudet omaavia ohjelmia kutsutaan englanninkielisellä termillä Rich Internet Application, lyhyemmin RIA. Termi voidaan suomentaa rikkaaksi internetsovellukseksi.

Ihanteellisessa MVC-mallissa (Model-View Controller) siirretään palvelimen ja asiakaskoneen välillä ainoastaan tietokannan raakadataa, joka käsitellään asiakaskoneella siihen ladatun sovelluksen toimesta. Tuloksena on siirrettävän datavirran määrän väheneminen siirtokaistalla ja näin ollen ohjelman nopeutuminen.

Eräänä Web 2.0:n määrittämisnä on visuaalisesti kehittyneemmät internetsivut, paremmat mahdollisuudet audion ja videon käyttöön sekä alustariippumattomuus (Web 2.0:n tietosivu Wikipediassa). Nämä tavoitteet ovat sellaisia, joihin RIA pystyy vastaamaan.

4.1 Web 2.0

Web 2.0:n tuotteet ja palvelut tuovat niitä käyttävälle selaajalle enemmän toiminnallisuutta ja enemmän mahdollisuuksia vaikuttaa palveluiden varsinaiseen sisältöön. Tällaiset sosiaaliset sovellukset ovat Web 2.0:n näkyvintä antia.

Web 2.0 on terminä hankala määrittellä ja sen käytöstä on hieman erimielisyyksiäkin, mutta edellämainitut enemmän toiminnalliset ja sosiaalisemmat verkkosovellukset ovat olennainen osa sitä.

4.2 Rich Internet Application (RIA)

RIA on internetissä toimivia ohjelma, joka on työpöytäohjelman kaltainen (RIA:n tietosivu Wikipediassa), runsaampi ja tietoturvaltaan kehittyneempi verrattuna client-server ajattelutapaan, jollaista puhdas HTML-perustainen internet-ohjelmointi on. Lisäksi oleellista RIA:ssa on kehitystyön helpottuminen verrattuna aikaisempaan esimerkiksi uudelleenkäytettävyyttä parantamalla.

RIA:n määritelmä ei sisällä teknisiä kohtia, vaan suuntaviivoja sille, mitä kyseisellä tekniikalla tulisi pystyä tehdä. RIA on sovellus, jota käytetään tavallisella verkkoselaimella, mutta sisältää ohjelmatason, jolla voidaan hoitaa sivun päivittäminen dynaamisemmin kuin tavanomaisella internetsivustolla. (RIA:n tietosivu Wikipediassa.)

RIA termi esiteltiin ensimmäistä kertaa Macromedian toimesta vuonna 2002 ja samantapaisia nimityksiä samalle asialle on ollut myös Microsoftilla ja Forrest Researchilla.

RIA-sovellus voi koostua joko pakatusta tiedostosta, joka sisältää toimintalogiikan (esimerkiksi SWF-tiedosto), tai paketista, jossa määritellään sivujen ulkoasu, kuvat ja näkyvän sivun toimintalogiikka, joka on toteutettu esimerkiksi Ajax-yhdistelmällä (Asynchronous JavaScript and XML).

4.3 Adobe Flex

Flex on alunperin Macromedian vuonna 2004 kehittämä ohjelmointikehys, joka mahdollistaa kehittyneiden internetohjelmien toteuttamisen riippumatta WWW-palvelimen alustasta. Flexin kehitys siirtyi Adobelle tämän ostettua Macromedian vuonna 2005 (Macromedian tietosivu Wikipediassa). Flexin ohjelmistokehityspaketti (SDK, Software Development Kit) on kuulunut Mozillan avoimen lisenssin alaisuuteen vuodesta 2008, kun Adobe julkaisi siitä version 3 (Adobe Flex -tietosivut Wikipediassa). Ohjelmistokehityspaketti on ladattavissa Flexin WWW-sivuilta. Ohjelmistokehityspaketti riittää pelkästään Flex-ohjelmien tekemiseen, mutta tehokkaaseen työskentelyyn tarvitaan kaupallinen Adobe Flex Builder -kehitysympäristö, joka on Eclipse-sovelluskehittimelle laadittu laajennus. Flex Builder -sovelluskehittäjästä on saatavilla vakaat versiot yleisimpiin käyttöjärjestelmiin kuitenkin sillä poikkeuksella, että Linux-ympäristöön on saatavilla ainoastaan ilmainen Alpha-versio, joka ei sisällä kaikkia samoja ominaisuuksia kuin maksulliset Mac- ja Windows-versiot.

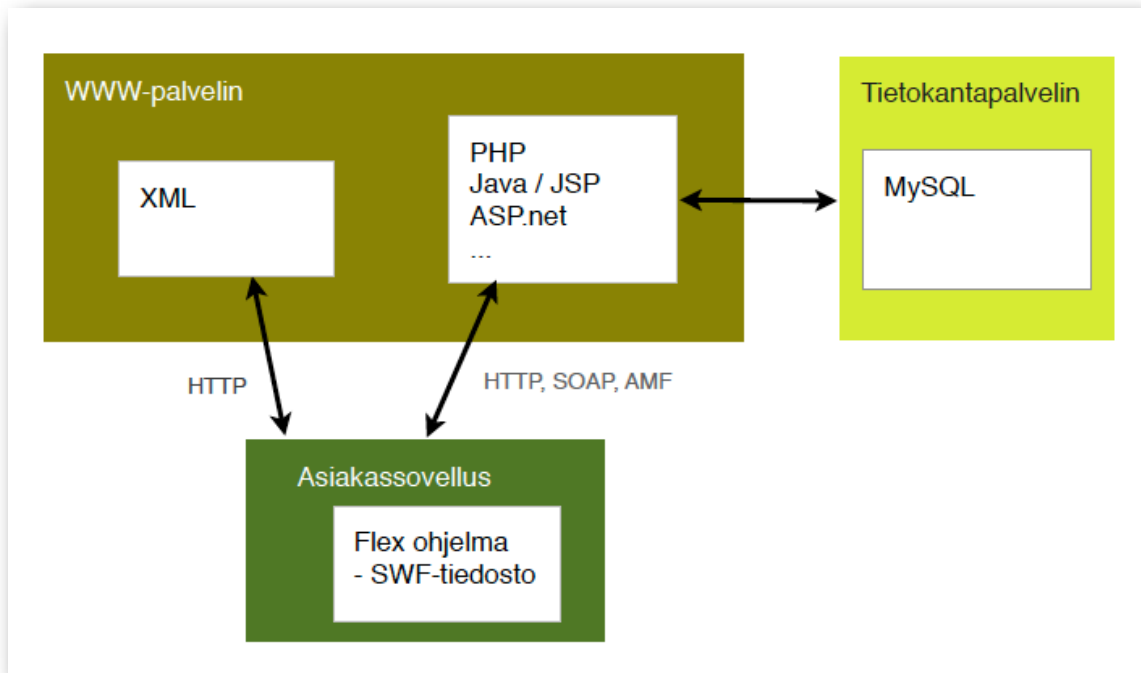
Flash-tekniikkaa ei ole suunniteltu laajojen ohjelmien tekemiseen ja näiden puutteiden tuomien kokemusten perusteella Flex:iä on kehitetty enemmän ohjelmistoteknisiin tarkoituksiin ottaen huomioon varsinkin insinööripohjaiset Java-ohjelmistosuunnittelijat (Kazoun & Lott 2007). Tämä ilmenee esimerkiksi käyttöliittymäsuunnittelun helppoudessa, samantyyppisessä kehitysrakenteessa kuin ASP.net:ssä (Active Server Pages) sekä helpoissa näyttökomponentteihin liitettävissä efekteissä.

Flex-ohjelman toiminta perustuu kolmeen osa-alueeseen: ActionScript 3 -ohjelmointikielen, XML-pohjaiseen (Extensible Markup Language) MXML-kuvauskieleen ja CSS -tyylimäärittelyihin (Cascading Style Sheets). MXML-kielellä kuvataan ohjelman rakenne ja käyttöliittymä. Flex sisältää useita valmiita

näyttökomponentteja, ja avoimmuun ansioista niiden muokkaaminen ja kehittäminen on mahdollista. Flexin ohjelmistokehityspaketin näyttökomponenttien lisäksi on saatavilla ilmaisia ja kaupallisia komponenttikirjastolaajennuksia, kuten FlexLibs ja ILOG Elixir.

Flex-sovelluksen toiminnot toteutetaan ActionScript-ohjelmointikielellä, jonka versionumero on 3. Flash-ohjelmissa käytettävät ActionScript 2 -kielellä toteutetut toiminnot eivät ole yhteensopivia Flexin kanssa eikä päinvastoin. MXML-näyttökomponenttien tyyliä määritellään CSS-tyylimäärittelyillä.

Flex-ohjelmakehityksen tuotantovaiheessa tehty ohjelmakoodi koostetaan rakennusvaiheessa SWF-tiedostoksi. Rakennusvaihe koostuu kääntämisestä, virheiden etsinnästä ja ohjelman testauksesta. Käännösvaiheessa kääntäjä kokoaa kaikki MXML-ohjelmakomponentit, ActionScript-lähdekoodit ja ulkopuoliset kirjastotiedot yhdeksi SWF-tiedostoksi, joka on valmis lähetettäväksi asiakaskoneen verkkoselaimelle avattavaksi. Käännösvaiheessa tulleet virheet ja varoitukset käsitellään ja ohjelman kääntäminen keskeytetään tarvittaessa. Ohjelman virheenetsintää varten Adoben sivuilta voi ladata Flash Player Debugger -version, joka tulostaa virheilmoitukset ja varoitukset ohjelman suorituksen aikana (run-time). Flex Builderissa on mahdollista määritellä tarkistuspisteitä ohjelman suoritukseen (breakpoints), joiden avulla ohjelman kulun tarkkailu on mahdollista ohjelman suorituksen aikana ja virheiden etsintä helpottuu. Flex-ohjelman toiminta nähdään kuvasta 1.



Kuva 1 Flexin toiminta

Tavallisessa käyttötapauksessa Flex-ohjelman toiminta on jaettu kolmeen osaan, joista näkyvimpänä on käyttäjän verkkoselaimessa toimiva SWF-tiedosto. Tämä ohjelma hakee tietokannan tiedot tietokantapalvelimelta WWW-palvelimella olevien toimintojen kautta. Tällaisessa toimintamallissa on se hyvä puoli, että tietokantapalvelinta voidaan vaihtaa, puuttumatta varsinaiseen Flex sovellukseen muuttamalla ainoastaan tietokantaa käsittelevää ohjelmaa.

4.4 ActionScript 3

Action Script 3 on uusien versio Macromedian alunperin kehittämästä ja julkaisemasta ohjelmointikielstä, joka esiteltiin ensimmäistä kertaa vuonna 1999 Flash 5 -version yhteydessä (ActionScriptin tietosivu Wikipediassa). Flash oli alunperin vektorigrafiikan esitysmuoto verkossa, josta se kehittyi aikajana-animoinnin ja yksinkertaisen ActionScriptin kautta interaktiiviseksi ohjelmointiympäristöksi, jolla on periaatteessa mahdollista tehdä monimutkaisiakin tietojärjestelmiä. Flashin ollessa kuitenkin graafinen työkalu, se ei sovellu mittaviin ohjelmistoprojekteihin kovinkaan hyvin. ActionScript-ohjelmointikieli sen sijaan kehittyi kohti laajempaa kokonaisuutta. Tämä kehityssuuntien eroaminen on

varmasti yksi syy siihen, että ActionScript-kielestä toteutettiin 3. versio, ja se liitettiin Flex-kehitysympäristöön yhdessä MXML-kuvauskielen kanssa.

ActionScript 3 julkaistiin kesällä 2006 yhdessä Flash Player 9 -version kanssa. Se käyttää Flash Player AVM2- virtuaalikonetta (VM, Virtual Machine), aiempien versioiden käyttäessä AVM1 -virtuaalikonetta. Flash Player 9 -ohjelma on ensimmäinen versio kyseisestä tuoteperheestä, joka sisältää molemmat virtuaalikoneet (ActionScriptin tietosivu Wikipediassa).

ActionScript 3 koostuu kahden tason tietotyypeistä, joista ylemmällä tasolla sijaitsee seitsemän perustyyppiä ja alemmalla tasolla 17 tietotyyppiä, (ActionScriptin tietosivu Wikipediassa), jotka toimivat koko Flex-viitekehyksen ytimenä. Kaikki Flexin näyttökomponentit pohjautuvat näihin tietotyyppiin.

4.5 MXML

MXML esiteltiin alunperin Macromedian toimesta vuonna 2004. Se on XML-pohjainen kuvauskieli ja toimii Flex-ohjelman rakenteen kuvaajana. Flex-viitekehysessä tarjottu MXML-nimiavaruus (xmlns, XML namespace) sisältää useita näyttökomponentteja, joiden muokkaaminen on mahdollista Flex-kehityksen avoimuuden ansiosta. Flex-ohjelman tulee sisältää ainakin yksi MXML-komponentti, jonka sisältö määritellään joko staattisesti suoraan komponenttiin tai dynaamisesti ActionScript-metodien kautta. MXML-tagit ovat itseasiassa ActionScript-luokkien ilmentymiä. Näyttökomponentti voidaan siis määritellä joko MXML-tageina tai ActionScript-luokkina. Käännösvaiheessa MXML-komponentit käännetään ensin ActionScript3-luokiksi, tämän jälkeen ABC-tavukoodiksi, jonka jälkeen niistä muodostetaan SWF-tiedosto (Patrick).

Verrattuna puhtaaseen ActionScript-ohjelmointiin MXML-näyttökomponenttien asettelu käyttöliittymään on helpompaa. Flex Builder 3 -työkalussa on Linux -versiota lukuunottamatta suunnittelumoodi, jossa komponenttien sijoittelu ruudulle tapahtuu WYSIWYG-pohjaisesti (What You See Is What You Get). Lisäksi toimintojen ja asettelun pitäminen erillään on mahdollista, kun käytetään ActionScript- ja MXML-kieltä yhdessä.

4.6 Flex-ohjelmien käyttö

Flexin Flexillä toteutetut ohjelmat käännetään SWF-tiedostoiksi, joiden suoritus tapahtuu yleensä asiakaskoneen Flash Player -laajennukset avulla HTML-sivuun liitettynä. Toinen käyttötapa on Flex-sovelluksen lähdekoodin kääntäminen työpöytäsovellukseksi. Tästä aiheesta lisää kappaleessa Adobe AIR (Adobe Integrated Runtime). SWF-tiedostoksi käännetyn ohjelman avaamiseksi tulee verkkoselaimen yhteyteen asentaa Flash Player -laajennus, jonka versionumero on vähintään 9. Flash Player 9 -laajennus on saatavilla kaikkiin yleisimpiin käyttöjärjestelmiin ja verkkoselaimiin. Flex ohjelma ei pysty käsittelemään tietokantaa suoraan, vaan se täytyy tehdä palvelinpuolen tarjoamalla palvelulla. Tämä selitetään tarkemmin kohdassa Tietokannan käsittely (luku 4.9).

Niin kuin aikaisemmin on mainittu, Flexillä toteutetut ohjelmat ovat käyttäjän verkkoselaimessa suoritettavia ohjelmia. Tällä tavalla ne eroavat osittain samantyyppisestä ASP.net:stä (Kazoun & Lott 2007).

4.7 Adobe AIR

Flex-sovelluksia kehitetään etupäässä internetohjelmiksi, joita käytetään asiakaskoneen verkkoselaimella. Tämä ei ole ainut tapa ohjelman käyttämiseen. Flex-ohjelma voidaan kääntää myös työpöytäsovellukseksi, jonka käyttö on mahdollista ilmaisen Adobe AIR -laajennuksen avulla. AIR-sovelluksilla pysytään käyttöjärjestelmän resursseja käyttämään tehokkaammin ja turvallisemmin.

Adobe AIR -tekniikalla Flex-, Flash- tai Ajax-sovelluksia voidaan kääntää käytettäväksi työpöydällä (Adobe AIR -tuotesivu).

4.8 JavaScript ja Ajax

Flex-ohjelmointikehyksessä on tehokkaat rajapinnat verkkoselaintietojen käsittelyyn. IBrowserManager-luokan avulla on mahdollista käsitellä verkkoselaimen osoitekentää sekä sivuotsikoita ja External-pakettiin kuuluvien luokkien avulla on mahdollista kutsua Flex-sovelluksen omaavalle HTML-sivulle määriteltyjä JavaScript-funktioita. Tällä tavalla Flex-sovellus voi navigoida itsensä ulkopuolella ja WWW-sivuja ei tarvitse kokonaisuudessaan toteuttaa Flexillä, vaan esimerkiksi vain tyylikkään sivumenu luominen riittää. Ajax:in käytöllä vähennetään turhaa tiedonsiirtoa palvelimen ja asiakaskoneen välillä ja sivun latautumiseen menevää aikaa samalla lyhennetään. Tämä on yksi Flexin ja Ajaxin yhteiskäytön eduista; WWW-sivuilla voidaan esittää näennäisesti reaaliaikaista tietoa, kun sivukomponentti voidaan päivittää automaattisesti eikä siihen tarvita käyttäjän päivitystoimenpidettä. Sivua voidaan päivittää taustaprosessin avulla, ja käyttäjä välttyy töksähtävältä sivupäivitykseltä.

Ajax ei sisällä mitään uutta, vaan sitä voidaan ajatella yhdistelmänä vanhaa. Tiedonsiirron kannalta merkittävässä asemassa on JavaScript-luokka XMLHttpRequest, joka hoitaa asynkronisen datan siirtämisen palvelimen ja asiakaskoneen välillä. Käytännössä tämä tarkoittaa sitä, että HTML-sivua ei tarvitse kokonaisuudessaan päivittää selaimella, jotta saataisiin palvelimelta uusin tieto käyttöön, vaan XMLHttpRequest voi hoitaa tietyn osan sivun päivittämisestä automaattisesti. Ajaxin Wikipedia-sivulla määritellään Ajax:n ajatus:

- Esitystapana käytetään XHTML- ja CSS-tekniikoita.
- DOM-standardia (Document Object Model) käytetään viittaamaan dynaamisesti HTML-sivun eri kohtiin.
- XML-kuvauskieltä käytetään datan käsittelyssä.
- JavaScript-kieltä tai muuta asiakaspuolen skriptikieltä käytetään edellisten yhdistävänä tekijänä.

Asiakaspuolen skriptikielen käytöllä saadaan käyttöliittymä dynaamisemmaksi, eikä sivua tarvitse ladata aina uudestaan, jotta esimerkiksi lomakkeeseen syö-

tetyt tiedot saataisiin validoitua. Heikkoutena tällaisessa käyttötapauksessa on verkkoselaimen Takaisin-painikkeen toimimattomuus, sillä dynaamisesti ladattu sisältö ei rekisteröidy selaimen historiatietoihin.

4.9 Tietokannan käsittely

Flex ei sisällä tietokannan käsittelyyn liittyviä ominaisuuksia, vaan yhteydet tietokantaan täytyy hoitaa jonkin palvelinpuolen sovelluksen välityksellä. Ohjelmakutsuun voidaan käyttää HTTP-kutsua (Hypertext Transfer Protocol) GET- tai POST-parametreillä, jolloin myös ulkopuolisella WWW-palvelimella olevia ohjelmia voidaan käyttää tietokantayhteyden muodostamiseen. Muina vaihtoehtoina ovat SOAP- (Simple Object Access Protocol) ja AMF-kutsut (Action Message Format), joiden avulla voidaan käyttää etäobjekteja sekä tehdä kutsuja Flexin WebService-komponentin välityksellä. HTTP-kutsua voidaan käyttää myös XML-tyyppisen tiedon hakemiseen palvelimelta.

Flex-ohjelman kehitys voidaan katsoa koostuvan suunnittelu-, konfigurointi-, rakennus-, järjestely- ja suojausvaiheista. Ohjelma voidaan suunnitella käyttäen MVC-ohjelmakehitysmallia esimerkiksi siten, että ohjelman käyttämä data sijoitetaan ulkopuoliseen XML-tiedostoon, joka ladataan HTTP-kutsuna Flex-ohjelmaan, ohjelman toiminnot määritellään ulkopuolisissa ActionScript-tiedostoissa ja ulkoasu koostetaan MXML-komponenteiksi.

Konfigurointivaiheessa varmistetaan, että ohjelmalla on mahdollisuus toimia siinä ympäristössä, johon se toteutetaan. Tähän liittyy tarvittavien datayhteyksien avaaminen. Tähän liittyvä määrittely tapahtuu muun muassa tiedostossa `crossdomain.xml`, jolla palvelin tunnistaa sellaiset dokumentit ja datapalvelut, jotka ovat SWF-tiedoston käytettävissä. Kuvassa 2 on esimerkki `crossdomain.xml` -tiedostosta.

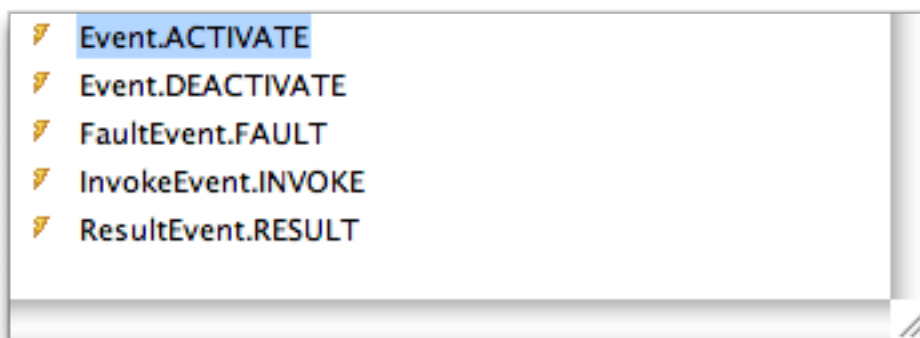
```
<?xml version="1.0"?>
<!DOCTYPE cross-domain-policy
SYSTEM "http://www.macromedia.com/xml/dtds/cross-domain-policy.dtd">
<cross-domain-policy>
    <allow-access-from domain="195.192.179.241" />
</cross-domain-policy>
```

Kuva 2 Esimerkki crossdomain.xml -tiedostosta

Tiedoston cross-domain-policy -tagien sisällä määritellään säännöt, joiden perusteella sallitaan Flex-sovelluksen yhteys palvelimella oleviin resursseihin. Flash Player, jonka päällä Flex-sovellus ajetaan, tarkistaa kohdepalvelimelta onko verkkotunnuksella oikeus käyttää kohdepalvelimen resursseja. Edellisessä esimerkissä on sallittu resurssikäsitteilypyynnöt ainoastaan IP-osoitteesta (Internet Protocol) 195.192.179.241. crossdomain.xml on sijoitettava kohdepalvelimen WWW-kansion juureen.

4.10 HTTPService

HTTPService-luokkaa käytettäessä Flex-sovellus käyttää yhteyden muodostamiseen HTTP-protokollaa. Metodien lähetyksen jälkeen tarkkaillaan HTTPService-luokan olion tapahtumatietoja (event). Tilatiedot on listattu kuvassa 3.



Kuva 3 HTTPService-luokan tapahtumat

Tärkeimpinä tapahtumina on RESULT ja FAULT, jotka käynnistyvät, kun HTTPService-olio palauttaa joko tuloksen palvelusta tai virheen. HTTPService-oliolle voidaan määritellä palautettavan datan tulosformaatti. Käytettävät tulosformaatit

ovat array, e4x, object, text, xml tai flashvars. Tulosformaatti e4x:n (ECMAScript for XML) avulla XML-tyyppisen datan käsittely on helppoa ja nopeaa.

4.11 XML

XML on yleinen kieli, jolla kuvataan datan lisäksi sen merkitystä ja rakennetta. XML-kieltä voidaan käyttää monissa yhteyksissä tiedonvälitykseen palveluiden välillä. XML eroaa HTML-kuvauskielestä sillä, että siihen ei määritellä datan esitystapaa lainkaan. (XML-kielen tietosivu Wikipediassa)

XML:ää voidaan käyttää myös ohjelmistoriippumattomana tiedon tallennustapana. Flexissä XML-kieltä voidaan käyttää tiedon siirtämiseen ja tallentamiseen, lista- taulukko- ja valikkokomponenttien lähteinä tai vektorigrafiikkakuvien esittämiseen (SVG, Scalable Vector Graphics). Kuvassa 4 on esitelty XML-tyyppinen muuttuja.

```
[Bindable] public var lista:XML = <menu>
  <rivi label='Ensimmäinen rivi' />
  <rivi label='Toinen rivi' />
  <rivi label='Kolmas rivi' />
</menu>;
```

Kuva 4 Malli XML-muuttujan määrittelystä ActionScript 3- ohjelmointikielellä.

Muuttujalla on kolme Rivi-nimistä tietojäsentä, joista jokaisella on yksi Label-niminen attribuutti. Oikeaoppinen XML-tiedosto sisältää vain yhden juuritagin, joka tässä tapauksessa on nimetty Menu-tagiksi. Toisin kuin HTML-kielellä, XML:llä ei ole yleistä rakennemäärittelytapaa (DTD, Document Type Definition), joka määrittäisi hyväksyttävien tagien nimet ja ominaisuudet, vaan nimet ovat suunnittelijan päätettävissä. XML-tietotyyppi voidaan sitoa MXML-näyttökomponentin lähteeksi. Kuvassa 5 on esitelty ComboBox-tyyppinen näyttökomenttirivi.

```
<mx:ComboBox id="combo1" dataProvider="{lista.rivi}" labelField="@label"/>
```

Kuva 5 ComboBox-tyyppinen näyttökomponenttirivi MXML-lähdekoodissa

Näyttökomponentin dataProvider-attribuutissa määritellään komponentin tietolähde, joka on esimerkissä lista-muuttujan Rivi-tietue. Attribuutissa labelField määritetään komponentissa esitettävä tieto. Merkillä @ viitataan attribuuttiin. ComboBox on esitetty kuvassa 6.



Kuva 6 ComboBox näyttökomponentti SWF-tiedostossa

Ohjelman kääntämisen jälkeen muodostettu SWF-tiedosto sisältää ComboBox-komponentin, jonka valintariveinä on Lista-muuttujan Rivi-tietueiden Label-attribuutit.

4.12 AMF

AMF on binäärimuotoinen dataformaatti, joka on suunniteltu erityisesti Flash-ohjelmien tiedonsiirtoa varten (AMF tietosivu Wikipediassa). AMF-muotoisen siirtodatan voi tallentaa suoraan ActionScript-muuttujaksi, jolloin se on nopeampaa kuin XML-muotoisen datan käsittely. Opinnäytetyön yhteydessä tehdyn testin perusteella 2500 tietueen mittaisen AMF-muuttujan käsittely tietokannasta Flex-sovelluksen DataGrid-näyttökomponenttiin oli yli neljä kertaa nopeampi kuin XML-tyyppisen muuttujan käsittely samalla datalla. DataGrid-näyttökomponentti on esitelty kuvassa 7. Myös DataGrid-komponentin järjestäminen oli AMF-tiedolla huomattavasti nopeampaa kuin XML-tyyppisellä datalla.

5 JÄRJESTELMÄN TOTEUTUKSEN VAIHEET

Opinnäytetyön suunnittelu ja toteuttaminen aloitettiin kesäkuun alussa 2008. Tätä ennen tutustuttiin pienimuotoisesti Mediakolmion toimenkuvaan ja käytettäviin tekniikoihin. Osa käytettävistä tekniikoista oli jo ennastaan tuttuja, mutta uutena aiheena oli Adobe Flex.

5.1 Esitutkimus

Esitutkimusvaiheessa tutustuttiin tarkemmin Mediakolmion liiketoimintaan ja tehtävien ja projektien läpivientiin sekä asiakasrajapinnan hoitamiseen. Juuri käyttöönotetusta Mekko II -toiminnanohjausjärjestelmästä kerättiin palautetta käyttäjähaastatteluilla sekä omien testikokemusten kautta. Esitutkimusvaiheessa kuva siitä, mitä oltiin tekemässä selkeni ja tätä projektin kokonaiskuva tarkentui ja opinnäytetyön rajauksen teko helpottui. Aikaisempien projektien kokemusten perusteella juuri rajaukseen kiinnitettiin huomiota, jotta projekti ei kasva yli siihen varattujen rajojen. Yhtenä osa-alueena opinnäytetyön alussa oli myös Applen Mac OS X -käyttöjärjestelmään tutustuminen.

5.2 Määrittely

Määrittelyvaiheessa aloitettiin esitutkimusvaiheessa kerättyjen tietojen perusteella järjestelmän suunnittelu ja tarkempi aikatauluttaminen. Määrittelyvaiheessa suunniteltiin ne tietotarpeet, joita opinnäytetyössä toteutettavassa järjestelmässä tarvittiin, ja tätä kautta muodostettiin tietokantakaaviot. Toiminnot liittyivät jo olemassa olevaan järjestelmään, joten käytössä olevan tietokannan muuttamiselta ei myöskään vältytty.

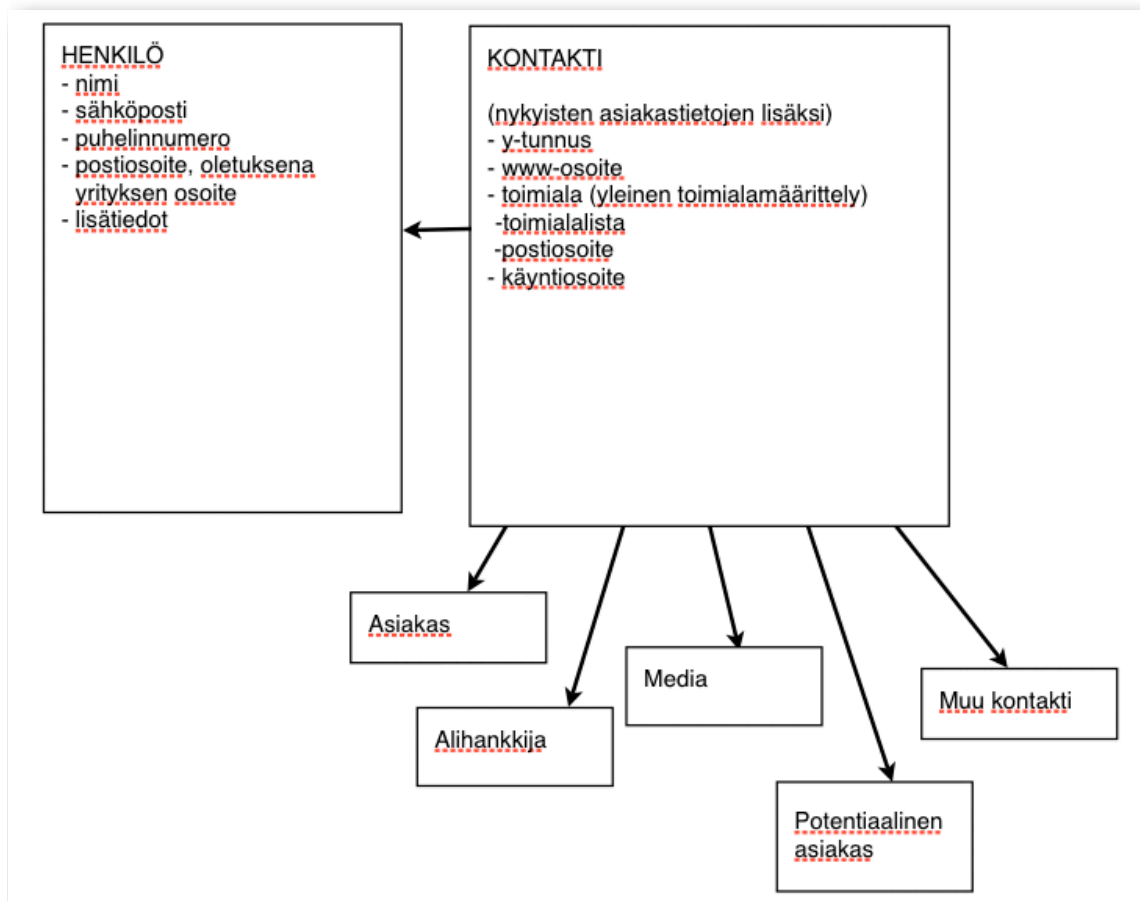
6 RATKAISU

Mediakolmio on kasvattanut henkilökuntaansa syksystä 2007 lähtien ja kasvu on aiheuttanut muutoksia projektinhallintaan sekä asiakkuuksiin ja seurantaan liittyviin toimiin. Hankittaessa uutta toiminnanohjausjärjestelmää Mediakolmion kaltaiseen yritykseen voidaan kärjistetysti ajatella, että vaihtoehtoja on kaksi: hankkia järjestelmä ulkopuoliselta taholta tai suunnitella ja toteuttaa järjestelmä itse. Johtuen aikaisemmista kokemuksista Mekon kehityksen suhteen sekä tästä johdetuista taloudellisista arvioista järjestelmän kehityksen tai ulkopuolisen hankinnan välillä, Mediakolmion tapauksessa valittiin jälkimmäinen vaihtoehto.

Opinnäytetyö koostuu kahdesta osasta: asiakastietokannan edelleenkehittämisestä kontaktitietokannaksi ja tehtävätrafiikin toteuttamisesta asiakaskäyttöliittymällä.

6.1 Kontaktitietotarpeiden kehittäminen

Nykyisessä järjestelmässä oleva asiakkaan käsittely muutetaan käsittämään asiakastietojen lisäksi alihankkijat ja mediat sekä potentiaaliset asiakkaat. Edellä mainituista käytetään nimitystä kontaktit. Nykyinen asiakastietokantataulu muutetaan kontaktitietokantatauluksi, johon on lisätty uudet kentät uusien tietotarvemäärittelyjen mukaisesti ja mahdollistettu kontaktin jakaminen asiakkaisiin, alihankkijoihin, potentiaalsiin asiakkaisiin ja muihin kontakteihin. Kontaktien tietotarpeet on esitelty kuvassa 8.



Kuva 8 Kontaktin ja siihen liittyvän henkilön tietotarpeet

Asiakkaalle on aikaisemmin ollut mahdollista tallentaa ainoastaan yksi osoitetieto. Tämä on aiheuttanut sen, että asiakkaan henkilöiden yhteystietojen ja roolien tallennus ei ole onnistunut. Toteutettavissa toiminnoissa mahdollistetaan henkilötietojen lisäämisen kontaktin yhteyteen ja yhteystietojen määrittelyn jokaiselle kontaktikohtaisesti. Henkilöön liittyvät tiedot ovat etunimi, sukunimi, henkilö-id ja henkilökohtainen lisätieto. Henkilön osoitetiedot määritellään henkilö- ja kontaktikohtaisesti, jolloin henkilön liittäminen useampaan kontaktiin eri osoitetiedoilla on mahdollista.

Taulukko 2 Kontaktien henkilöiden käyttöoikeudet Mekossa

KÄYTTÄJÄOIKEUS	SELITE
Ei Mekko tunnuksia	Henkilöllä ei ole oikeutta nähdä kyseisen kontaktin tehtäviä ja projekteja.
Selaaja	Henkilö voi kirjautua Mekkoon ja tarkastella tehtäviä, projekteja, vedoksia ja julkisia kommentteja, jotka ovat häneen liittyvän kontaktille lisätty.
Kommentoija	Henkilö voi kirjautua Mekkoon ja tarkastella tehtäviä ja projekteja, jotka ovat häneen liittyvän kontaktille lisätty sekä lisätä kommentteja tehtäville.

Mikäli henkilöllä ei ole yhtään Selaaja- tai Kommentoija-tason käyttöoikeutta, ei hän voi kirjautua Mekkoon, vaikka hänelle olisikin määritelty käyttäjätunnus ja salasana. Kommentointi ja tehtävätrafiikin toiminnot on kuvattu tarkemmin luvussa 6.2.

6.2 Tehtävätrafiikki

Mekkoon tallennetaan tehtävät alkaen niiden luonnostelusta aina laskutusvaiheeseen asti. Laskutuksen jälkeen tehtävät jäävät arkistoon tietokantaan. Tehtävälle lisätään työmerkintöjä yhdestä useaan kymmeneen riippuen sen laajuudesta ja luonteesta.

Mekkoon toteutetaan asiakasliittymä, jonka kautta asiakkaan henkilön on mahdollista tarkastella omia tehtäviään, niiden valmiusastetta, kommentteja sekä vedostiedostoja.

Kommentilla voi olla tiedosto, tai se voidaan tallentaa tietokantaan ilman sitä. Sallitut tiedostotyypit tallennetaan tauluun Tiedostotyypit.

7 YHTEENVETO

Opinnäytetyöni aiheena oli lappeenrantalaisen mainostoimiston toiminnanohjausjärjestelmään liittyvä kehitystyö, jolla pyrittiin tuomaan toiminnanohjaamisen kehittämisen lisäksi lisäarvoa toimiston asiakkaille. Työn aikana opin, että yrityksen toiminnanohjausjärjestelmä on todella merkittävä osa yrityksen toimintaa. Toiminnanohjausjärjestelmällä on kaksi tarkoitusta; auttaa työntekijöitä omien työtehtävien priorisoinnissa ja yleensäkin työn suorittamisessa sekä tuoda yrityksen johdolle arvokasta tietoa siitä, mikä on yrityksen tilauskannan koko ja miten yritys toimii verrattuna sille asetettuihin tavoitteisiin.

Työssäni käyttämistä työkaluista ja toimintatavoista olen oppinut paljon työn aikana. Uskon, että tästä opista on paljon hyötyä työuraani ajatellen. Ohjelma-
tuotanto on muuttunut paljon viime vuosina. Tänä päivänä lähtökohdat ohjelmointityölle ovat erilaisissa ohjelmointityötä helpottavissa viitekehyksissä, jolloin ohjelmoijan tehtävänä ei ole välttämättä osata ohjelmointikieltä läpikotaisin, vaan osata käyttää kehitystyöhön toteutettuja työkaluja mahdollisimman tehokkaasti.

Myös internetohjelmointi on kehittynyt ja kehittyy edelleen. Perinteisestä palvelin-asiakasmallista on siirrytty eteenpäin sovellusajatteluun, jossa kokonaisia, aikaisemmin työpöytäohjelmina suoritettuja toimintoja voidaan siirtää toimimaan internetissä. Tästä voidaan mainita esimerkkinä Google Docs -toimistoohjelmat.

Uuden ohjelmointikielen lisäksi sain oppia paljon työskentelytavoista ohjelmistoprojekteissa ja siitä, kuinka tärkeä suunnitteluvaihe tämältyyppisissä töissä on. Ohjelmistoprojektia voisi verrata talon rakentamiseen; jos suunnitelmia muutetaan tuotantovaiheessa, aikataulu venyy helposti erittäin paljon ja työmäärä kasvaa rajusti.

Avoimen lähdekoodin ohjelmistot olivat aikaisemmin ainoastaan ohjelmointiin perehtyneiden harrastajien kiinnostuksen kohteena, mutta tänä päivänä siirrytään enemmän suuntaan, jossa myös kaupallisessa mielessä voidaan aja-

tella avoimen lähdekoodin mahdollisuuksia. Tämä muuttaa liiketoimintamalleja ohjelmistotuotantoon keskittyvässä liiketoiminnassa.

Organisaation mallien mukaan oikein toimivalla toiminnanohjausjärjestelmällä tuotetaan projektien läpivientiä ja lisäarvona tuodaan asiakasrajapinta lähemmäksi tuotantoa, pitäen asiakas ajantasalla projektien ja niihin liittyvien tehtävien tilanteesta. Tämä toimintatapa vahvistaa toimittaja-asiakassuhdetta.

Kun etsin itselleni opinnäytetyön aihetta, tavoitteenani oli löytää työ, josta on hyötyä itselleni sekä työn tilaajalle ja joka on myös haastava. Nämä tavoitteet molemmat täyttyivät työssäni.

KUVAT

Kuva 1 Flexin toiminta, s. 17

Kuva 2 Esimerkki crossdomain.xml -tiedostosta, s. 22

Kuva 3 HTTPService-luokan tapahtumat, s. 22

Kuva 4 Malli XML-muutujan määrittelystä ActionScript 3 -ohjelmointikielellä,
s. 23

Kuva 5 ComboBox-tyyppinen näyttökomponenttirivi MXML-lähdekoodissa, s. 24

Kuva 6 ComboBox näyttökomponentti SWF-tiedostossa, s. 24

Kuva 7 Tyhjä DataGrid-näyttökomponentti, s. 25

Kuva 8 Kontaktin ja siihen liittyvän henkilön tietotarpeet, s. 28

TAULUKOT

Taulukko 1 Mekon käyttäjäryhmät, s. 12

Taulukko 2 Kontaktien henkilöiden käyttöoikeudet Mekossa, s. 29

LÄHTEET

ActionScriptin tietosivu Wikipediassa <http://en.wikipedia.org/wiki/ActionScript> (Luettu 23.3.2009)

Adobe AIR -tuotesivu <http://www.adobe.com/products/air/develop/Flex/> (Luettu 11.3.2009)

Adobe Flex Gumbo -tuotesivu <http://opensource.adobe.com/wiki/display/Flexsdk/Gumbo> (Luettu 8.3.2009)

Adobe Flex -tietosivu Wikipediassa http://en.wikipedia.org/wiki/Adobe_Flex (Luettu 14.10.2008)

Ajax-tietosivut Wikipediassa [http://en.wikipedia.org/wiki/Ajax_\(programming\)](http://en.wikipedia.org/wiki/Ajax_(programming)) (Luettu 10.3.2009)

AMF:n tietosivu Wikipediassa http://en.wikipedia.org/wiki/Action_Message_Format (Luettu 12.3.2009)

Kazoun, C & Lott, J. 2007. Programming Flex 2 2007. O'Reilly Media Inc.

Patrick, Ted: MXML vs. AS3 http://www.onFlex.org/download/mxml_vs_as3.pdf (Luettu 1.3.2009)

Macromedian tietosivu Wikipediassa <http://fi.wikipedia.org/wiki/Macromedia> (Luettu 13.10.2008)

Mediakolmion WWW-sivut <http://www.mediakolmio.fi> (Luettu 12.10.2008)

RIA:n tietosivu Wikipediassa http://en.wikipedia.org/wiki/Rich_Internet_application (Luettu 23.3.2009)

Web 2.0:n tietosivu Wikipediassa http://fi.wikipedia.org/wiki/Web_2.0 (Luettu 11.11.2008)

XML-kielen tietosivu Wikipediassa <http://fi.wikipedia.org/wiki/XML> (Luettu 10.3.2009)