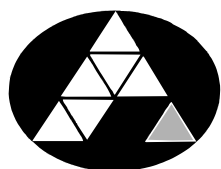


POHJOIS-KARJALAN AMMATTIKORKEAKOULU  
Viestinnän koulutusohjelma

Tero Hyttinen

JAVA ME:N JA WRT:N SOVELTUVUUS GPS-  
SISÄLTÖPALVELUUN

Opinnäytetyö  
Huhtikuu 2011



POHJOIS-KARJALAN  
AMMATTIKORKEAKOULU

**OPINNÄYTETYÖ**  
**Huhtikuu 2011**  
**Viestinnän koulutusohjelma**

Länsikatu 15  
80200 JOENSUU  
p. (013) 260 6990

Tekijä(t)  
Tero Hyttinen

Nimeke  
Java ME:n ja WRT:n soveltuvuus GPS-sisältöpalveluun

Toimeksiantaja  
Pohjois-Karjalan ammattikorkeakoulu, TG4NP-hanke.

**Tiivistelmä**

Opinnäytetyö käsittelee Java ME- ja WRT-ohjelmointiympäristöjen soveltuvuutta paikkatietoon sidonnaisen sisältöpalvelun toteutukseen. Opinnäytetyössä käsitellään opastuspalveluita laajennetun todellisuuden palveluiden näkökulmasta. Työn tavoitteena oli valittuja ohjelmointiympäristöjä vertailemalla löytää teknisiltä ominaisuuksiltaan paremmin palvelumuotoon soveltuva vaihtoehto. Vertailun lähtökohdaksi työssä tutkittiin laajennetun todellisuuden palveluiden määritteitä sekä aiemmin Pohjois-Karjalassa toteutetuissa laajennetun todellisuuden palveluissa esiintyneitä ongelmia.

Työn toiminnallinen tavoite oli toteuttaa kaksi matkapuhelinsovellusta valittuja ohjelmointiympäristöjä käyttämällä. Matkapuhelinsovellukset toimivat laajennetun todellisuuden palveluina tarjoten ääni- ja kuvasisältöjä. Sovelluksissa käytettiin paikkatietoa yhdistämään tarjotut sisällöt opastuskohteisiin. Palvelun kohdeympäristönä toimi Lieksassa sijaitseva Vuonislahden kylä. Työssä vertailtiin toteutettuja matkapuhelinsovelluksia sekä tarkasteltiin niiden soveltuvuutta laajennetun todellisuuden palvelualustoiksi.

Vertailun tulokset osoittivat, että WRT oli parempi vaihtoehto laajennetun todellisuuden palvelualustaksi. Vaikka Java ME toimi WRT:tä useammassa matkapuhelinmallissa, on WRT käytävyyden ja suorituskyvyn kannalta parempi vaihtoehto.

Työssä käsiteltiin matkapuhelinsovelluksia kirjoittajan omien kokemusten ja taustatiedon pohjalta. Lähdeaineistona työssä käytettiin laajennetun todellisuuden palveluita käsittelevää kirjallisuutta, internetlähteitä sekä aiemmin toteutettuja palveluita. Toiminnallisena lopputuloksena syntyneet sovellukset jäävät pilottitoteutuksina Pohjois-Karjalan ammattikorkeakoulun TG4NP-hankkeen testaus- ja tutkimuskäyttöön.

Kieli  
suomi

Sivuja 56  
Liitteet 5  
Liitesivumäärä 7

**Asiasanat**

GPS, HTML, Java ME, laajennettu todellisuus, matkapuhelinsovellus, paikkatieto, WRT, XML



NORTH KARELIA  
UNIVERSITY OF APPLIED SCIENCES

**THESIS**  
**April 2011**  
**Degree programme in**  
**Communications**  
Länsikatu 15  
FIN 80200 JOENSUU  
FINLAND  
Tel. 358-13-260 6990

Author(s)  
Tero Hyttinen

Title  
Java ME and WRT's Suitability for GPS Content Service

Commissioned by  
North Karelia University of Applied Sciences, TG4NP-project

#### Abstract

This bachelor's thesis deals with the suitability of Java ME and WRT-software development environments in implementing location based mobile phone content service. The thesis deals with the guidance services from the perspective of augmented reality services. The goal of this thesis was to find a better software development environment for the service by comparing technical features of selected environments. As a baseline for comparison the thesis used the main features of augmented reality services and examined development needs and problems from previously carried out services in North Karelia.

The functional purpose of the thesis was to create two augmented reality mobile phone applications by using selected software development environments. The created applications work as augmented reality mobile phone services and contain audio and visual contents. Mobile applications used the location based services to combine the contents to the real world places. The service was located in Vuonismaa-village in Lieksa, North Karelia. The thesis compared the created mobile applications and examined their suitability as a platform for augmented reality services.

The result of the comparison was that WRT software development environment found to be a better option to be used as a platform for augmented reality service. Even though Java ME-environment supports more devices, the WRT was better both in usability and performance.

The thesis used the author's previous experiences and background knowledge of mobile phone applications as a source. Other used sources were professional literature about augmented reality services, internet sources and knowledge from previously carried out services. The applications will remain with NKUAS TG4NP project for testing and research purposes.

Language  
Finnish

Pages 56  
Appendices 5  
Pages of Appendices 7

Keywords  
augmented reality, GPS, HTML, Java ME, location based services, mobile phone application, WRT, XML

# SISÄLTÖ

## TIIVISTELMÄ

## ABSTRACT

LYHENTEET JA TERMIT .....	6
1 JOHDANTO .....	8
2 LAAJENNETUN TODELLISUUDEN OPASTUSPALVELU .....	9
2.1 Palvelumuodon määrittely .....	9
2.2 Paikkatietoon sidonnaisen sisällön toimintaperiaate .....	9
2.3 Realismin rikastaminen matkapuhelinpalvelulla .....	10
3 MOBIILIOASTUKSET POHJOIS-KARJALASSA .....	12
3.1 Rajaus .....	12
3.2 GPS-Taival .....	12
3.3 Turusen taival .....	14
3.4 Palveluiden jatkokehitys .....	14
4 VUONISLAHDEN KYLÄKÄVELY .....	16
4.1 Palvelun suunnittelu .....	16
4.2 Sovelluskehityksen lähtökohdat .....	16
4.3 Ohjelmointiympäristöt .....	17
4.3.1 Rajaus .....	17
4.3.2 Java ME .....	18
4.3.3 WRT .....	19
4.4 Toteutuksen suunnittelu ja esivalmistelut .....	21
4.4.1 Tavoite .....	21
4.4.2 Käyttöliittymäsuunnittelu .....	21
4.4.3 GPS-mittaukset .....	23
4.4.4 Sisältömateriaalit .....	24
4.5 Java ME -sovellusversio .....	25
4.5.1 JM-Mobile Editor .....	25
4.5.2 Sovelluspohjan määrittely .....	25
4.5.3 Verkkopohjainen XML-rakenne .....	27
4.5.4 Sisältömateriaalien toistaminen .....	30
4.5.5 Paikkatiedon yhdistäminen sisältöihin .....	31
4.5.6 Sovelluksen ulkopuoliset sisällöt .....	33
4.5.7 Sovelluksen julkaisu .....	33
4.6 WRT-sovellusversio .....	34
4.6.1 Lähtötilanne .....	34
4.6.2 Sovelluksen rakenne .....	34
4.6.3 Ulkoasu .....	36
4.6.4 Platform Services API .....	37
4.6.5 Sisältömateriaalien toistaminen .....	39
4.6.6 Julkaisu ja jakelu .....	40
5 TULOKSET .....	42
5.1 Soveltuvuus .....	42
5.2 Prosessi .....	43
5.3 Työkalut .....	43
5.4 Rakenne .....	44
5.5 Suorituskyky .....	45
5.6 Sisältömateriaalin toistaminen .....	46
5.7 Ulkoasu .....	46

5.8	Ominaisuudet.....	47
5.9	Yhteensopivuus .....	48
6	POHDINTA .....	50
6.1	Tulosten yhteenveto .....	50
6.2	Vertailun vaihtoehtoinen tulos.....	51
6.3	Käytetyt menetelmät .....	52
6.4	Opinnäytetyöni oppimisprosessina .....	52
6.5	Palveluiden tulevaisuus ja jatkokehitysideat .....	53
	LÄHTEET.....	55

#### LIITTEET

Liite 1	Vuonislahden kyläkävelyn rakennekaavio
Liite 2	XML-pohjaisen sovellusrakenteen toimintaperiaate
Liite 3	WRT-sovellusversion kokoonpano
Liite 4	Java ME -sovellusversion ulkoasu
Liite 5	WRT-sovellusversion ulkoasu

## LYHENTEET JA TERMIT

A-GPS	Assisted GPS, avustettu satelliittipaikannus, on satelliittipaikannusta ja jotain avustetta yhdistävä paikannuskeino. Yleensä avusteen satelliittipaikannukselle antaa matkapuhelinverkko.
API	Application programming interface eli rajapinta on kokoelma erilaisia komentoja, joiden avulla sovellukset voivat tehdä pyyntöjä päätelaitteelle ja vaihtaa tietoja päätelaitteen ja sen muiden sovellusten kanssa. Rajapinnan avulla sovellukset voivat esimerkiksi käyttää puhelimesta olevia sensoreita, kuten GPS-sirua ja puhelimen asentoa seuraavia kiihtyvyyssensoreita.
Bluetooth	Avoin lähietäisyydellä toimivan langattoman tiedonsiirron standardi, joka perustuu radioteknologiaan eikä vaadi esteetöntä näköyhteyttä kommunikoivien laitteiden välillä.
CSS	Cascading Style Sheets, verkkosivuilla ja verkkosovelluksissa käytetty kieli, jolla kuvataan visuaalista ulkoasua ja asettelua. Kielen tarkoituksena on erottaa kohteen muotoilu sen sisällöstä.
GPS	Global Positioning System on Yhdysvaltojen puolustusministeriön kehittämä ja ylläpitämä satelliittipaikannusjärjestelmä.
HTML	Hypertext Markup Language on verkkosivuilla ja verkkosovelluksissa käytetty kieli, jolla kuvataan verkkosivujen rakennetta ja sisältöä.
JAD	Java ME -sovelluksien jakelutiedosto, joka sisältää kaiken sovelluksessa olevan paikallisen datan.
JAR	Java ME -sovelluksien määrittelytiedosto, joka sisältää sovelluksen perustiedot, kuten nimen ja toteuttajan, sekä asennuksen yhteydessä sovellukseen asetettavat asetukset, kuten käytettävän näyttömoodin.
JavaScript	Oliopohjainen asiakassovelluksissa ja verkkosivuilla käytetty komentosarjakieli, jota käytetään esimerkiksi verkkosivuilla lisäämään dynaamista toiminnallisuutta sisältöön.
PLIST	WRT-sovelluksissa käytetty sovelluksen määrittelytiedosto kertoo sovelluksen perustiedot, kuten nimen, toteuttajan ja rakennetiedoston.

- RAM** Random Access Memory eli käyttömuisti on tietokoneissa ja matkapuhelimissa oleva työmuisti, johon ladataan suoritettavien sovellusten tarvitsemat tiedot ja sisällöt. Käyttömuistiin lataamisen takia sovellukset ja niiden sisällöt toimivat nopeammin, koska jatkuva sisällön lataamisen tarve vähenee.
- WGZ** Nokian WRT-sovelluksien julkaisussa käytetty tiedostoformaatti on pakattu tiedostomuoto, joka sisältää kaiken sovelluksessa olevan paikallisen datan.
- WLAN** Wireless Local Area Network on langaton lähiverkko, jonka avulla erilaiset verkkoyhteydellä varustetut laitteet voivat olla toisiinsa yhteydessä sekä toistensa kautta yhteydessä esimerkiksi internetiin.
- WYSIWYG** What You See Is What You Get eli saat mitä näet on ohjelman visuaalista käyttöliittymää kuvaava termi.

# 1 JOHDANTO

Mobiiliteknologia on huimaa vauhtia kehittyvä ala, ja laitteiden jatkuvan teknisen kehityksen myötä niille tarjotut sisällöt ovat jääneet teknologiasta jälkeen. 2000-luvun alkupuolen uusmedia-alan finanssikupla ja sitä seurannut lama jättivät jälkeensä varovaisuutta, joka vaikuttaa sisällöntuotantopalveluissa vielä tänäkin päivänä (Kuivakari 2006, 7). Matkapuhelimet ovat kehittyneet ominaisuuksiltaan monitoimisiksi mobiilitietokoneiksi. Matkapuhelinten ominaisuuksien hyödyntäminen tarjottavissa sisällöissä on vähitellen parantumassa nopeasti kasvavan sovelluskehittelyn ja -kaupan ansiosta. (Vesa 2010.)

Matkapuhelinteknologian hyödyntäminen matkailupalveluissa on Suomessa toistaiseksi melko vähäistä. Maailmalla on muutamien vuosien ajan kehitetty matkailupalveluita asiakkaille vuokrattaville päätelaitteille. Jatkuvasti kehittyvän teknologian ansiosta palveluita pystytään tuomaan lähemmäksi käyttäjää toteuttamalla palvelut hänen omaan henkilökohtaiseen matkapuhelimeensa. Entuudestaan tuttu päätelaite madaltaa kynnystä virtuaalisten palveluiden käyttöön. (Ristolainen 2007, 3–12.)

Opinnäytetyössäni käsittelen Java ME ja WRT-ohjelmointiympäristöjen soveltuvuutta paikkatietoon sidonnaisen sisältöpalvelun toteutukseen. Tutkin sisältöjä ja paikkatietoja yhdistelevää opastuspalvelua laajennetun todellisuuden (augmented reality) näkökulmasta. Toiminnallisessa osuudessa toteutan kaksi teknistä pilottisovellusta valittuja ohjelmointiympäristöjä käyttämällä. Lisäksi vertailen toteutettujen sovellusten teknisiä ominaisuuksia ja soveltuvuutta paikkatietoa ja sisältömateriaaleja käyttävän opastuspalvelun alustaratkaisuksi. Käytän sovelluksien suunnittelun ja kehitystyön taustalla aiemmin Pohjois-Karjalassa toteutettuja mobiiliopastuksia ja niissä ilmenneitä ongelmakohtia. Toiminnallisena lopputuloksena syntyvissä sovelluksissa käytetään monikansallisen TG4NP-hankkeen<sup>1</sup> toteuttamaa Vuonislahden kyläkävely -pilotin sisältömateriaalia.

---

<sup>1</sup> Tourist Guide for Northern Periphery, Pohjois-Karjalan ammattikorkeakoulu.



## **2 LAAJENNETUN TODELLISUUDEN OPASTUSPALVELU**

### **2.1 Palvelumuodon määritteet**

Käsittelen opinnäytetyössäni paikkatiedon avulla tiettyyn sijaintiin yhdistettävää sisältöpalvelua. Kun sisältö yhdistetään johonkin todelliseen ympäristöön, on kyse realismin rikastamisesta eli laajennetusta todellisuudesta. Laajennetulla todellisuudella tarkoitetaan todellisuudessa tapahtuvan toiminnan monipuolistamista ja elävöittämistä teknologian keinoin (Kuivakari 2006, 80–81). Mobiililaitteilla laajennettua todellisuutta tavoitellaan tuomalla palvelun käyttämiseen uusia tapoja, kuten esimerkiksi sijaintiin ja käyttäjän liikkumiseen reagoivia sisältöjä. (Ylä-Kotola & Arai 2000, 54–55).

Matkapuhelimessa toimivalla laajennetun todellisuuden opastuspalvelulla tarkoitetaan sovellusta, jonka avulla käyttäjä pystyy eläytymään monimediaiseen sisältöön ja ymmärtämään sisällön yhteyden ympäröivän todellisuuden kanssa. Toisin kuin virtuaalitodellisuuksissa, laajennetun todellisuuden tavoitteena ei ole rakentaa erillisiä lumemaailmoja vaan enemmänkin rikastuttaa ympärillä olevaa realismia (Kuivakari 2006, 67). Realismia rikastava elementti voi olla esimerkiksi päätelaitteen kautta kuultava audio, joka kertoo ympärillä olevasta todellisuudesta jotain sellaista, minkä hahmottaminen on muuten käyttäjälle mahdotonta. Laajennetun todellisuuden vaikutelman tehostamiseksi sovelluksen sisältö voidaan asettaa tiettyyn fyysiseen sijaintiin toistuvaksi paikkatiedon avulla, jolloin kyseessä on motorisesti käytettävä sovellus. (Ylä-Kotola & Arai 2000, 54–55.)

### **2.2 Paikkatietoon sidonnaisen sisällön toimintaperiaate**

Paikkatietoon sidonnaiset mobiilipalvelut ovat motorista vuorovaikutusta käyttäjän ja käytettävän palvelun välillä. Vuorovaikutus perustuu käyttäjän toimintaan ja hänen fyysiseen liikkumiseensa opastuksen kohteessa. Paikkatieto mahdollistaa tämän vuorovaikutuksen käyttäjän liikkumisen ja tarjottavan sisällön välillä (Ristolainen 2007, 20).

Paikkatieto on käyttäjän fyysisen sijainnin, paikannusmenetelmän ja päätelaitteen välistä kommunikaatiota. Yleisin paikannusmenetelmä on GPS, joka on

Yhdysvaltojen puolustusministeriön kehittämä ja ylläpitämä satelliittipaikannusjärjestelmä. (Kowoma 2009.) Muita mobiililaitteilla käytettyjä paikannusjärjestelmiä ovat matkapuhelinverkkoa paikannusapuna käyttävä A-GPS sekä paikalliseen verkkoperustaiseen paikannukseen perustuvat WLAN- ja Bluetooth-menetelmät. Uusimmat matkapuhelimet yhdistelevät näitä eri menetelmiä saavuttaakseen tarkimman mahdollisen tuloksen sijainnin määrittämisessä. (Djuknic & Richton 2001.)

Paikkatietoon perustuva laajennetun todellisuuden mobiilipalvelu käyttää hyödyksi paikannusjärjestelmien tuottamaa tietoa käyttäjän sijainnista ja yhdistää mediasisällön sijainnin koordinaatteihin (Kuivakari 2006,67). Opastuspisteiden ympäriltä mitataan koordinaattipisteet eli niin sanotut raja-arvot, joiden sisälle muodostuu opastuspisteen toiminnallinen vaikutusalue (ks. kuvio 3). Kun käyttäjä liikkuu mediasisältöön asetettujen sijaintien raja-arvojen sisälle, tulee sijaintiin liitettyyn informaatioon tutustumisen mahdolliseksi päätelaitteella.

### **2.3 Realismin rikastaminen matkapuhelinpalvelulla**

Laajennetun todellisuuden palvelut pohjautuvat vuosisatoja vanhaan simulaatiokulttuuriin. Jo 1800-luvulla ihmisille tarjottiin simulaatioihin ja illuusioihin perustuvia elämyksiä panoraama- ja dioraama-esitysten avulla. Vaikka esitykset erosivat teknisesti nykypäivän lumetodellisuuden tavoittelusta, oli niiden päämääränä sama realismin rikastaminen. Panoraama- ja dioraama-esityksissä elämyksellisyyttä tavoiteltiin pitämällä mekanismit mahdollisimman piilotettuina ja tuomalla niiden kannattelema sisältö pääosaan esityksessä. (Kuivakari 2006,61–66.)

Kuivakarin (2006, 76) mukaan mobiililaitteille toteutetuissa opastuspalveluissa käyttäjä kokee sisällön kahdella tasolla liikkuvan subjektiivisuuden mukaan. Käyttäjä kokee sisällön joko illusorisesti – eli elämyksenä – tai todellisuutta täydentävänä oppimistehtävänä. Kuivakarin päätelmää soveltaen matkapuhelinten opastuspalvelut voidaan jakaa karkeasti kahteen kategoriaan: laajennetun todellisuuden opastuspalveluihin ja informatiivisiin opastuspalveluihin. Ero näiden kahden palvelumuodon välille tulee sisältöjen ja niiden herättämien reaktioiden kautta. Laajennetun todellisuuden palveluissa käyttäjälle tarjotaan realis-

mia rikastavaa sisältöä mahdollisimman arkipäiväisiin toimintoihin liitettynä. Laajennetun todellisuuden palveluissa käyttäjälle pyritään luomaan illuusio sisällön ja realismin välille ilman esiintuovien mekanismien tietoisuutta. Tämä illuusio voi olla esimerkiksi ajallinen, jolloin ympärillä olevaan todellisuuteen tuodaan siihen liittyvää historiallista informaatiota ja viedään käyttäjä sisällön avulla nykyhetkestä ajassa taaksepäin. (Kuivakari 2006, 67.)

Informatiivisessa opastuspalvelussa sisällöt esiintuova mekanismi on selkeämmin osana palvelua ja palvelun tavoitteena on täydentää todellisuutta tarjotulla informaatiolla. Informatiivisissa opastuspalveluissa sisältö tarjotaan käyttäjälle opetuksellisena informaationa ja sisällön esiintuovat mekanismit ovat keskeinen ja näkyvä osa toimintaa. (Kuivakari 2006, 65–67.)

## 3 MOBIILIOPASTUKSET POHJOIS-KARJALASSA

### 3.1 Rajaus

Mobiililaitteille suunnattuja opastuspalveluita on Suomessa tehty noin kymmenkunta (Ristolainen 2007, 3–12). Toteutetut palvelut ovat pääosin informatiivisia, enkä sen vuoksi tarkastele niitä sen tarkemmin. Keskityn käsittelemään ainoastaan Pohjois-Karjalassa toteutettuja palveluita, sillä olen niissä itse ollut opintojeni kautta yhtenä toteuttavana osapuolena. Esiteltävät palvelut ovat kaikki monikansallisen CMC@NP<sup>1</sup> -hankkeen ja Pohjois-Karjalan Maakuntaliiton johtamia pilottitoteutuksia, joissa on testattu virtuaalisten matkailupalveluiden käyttömahdollisuuksia matkailukohteiden esittelyssä. Palveluiden toteutuksesta on vastannut Pohjois-Karjalan ammattikorkeakoulu, ja oma tehtäväni on ollut vastata niissä käytetyistä teknisistä ratkaisuista.

### 3.2 GPS-Taival

GPS-Taival<sup>2</sup> on ensimmäinen Pohjois-Karjalassa toteutettu elämyksellinen mobiilipalvelu. Se kertoo Ilomantsin Hattuvaarassa käydystä jatkosodan taistelusta, ja sen kohteina ovat taisteluun osallistuneiden veteraanien tekemä Taistelijan Taival -luontopolku sekä veteraanien kertoma tarina taistelun etenemisestä. GPS-Taival koostuu kolmesta palvelumuodosta: internetsisällöstä, Google Maps -sisällöstä ja paikan päällä käytettävästä mobiilipalvelusta. Käsittelen opinnäytetyössäni GPS-Taipaleesta ainoastaan sen mobiilipalvelua. GPS-Taipaleessa käytettiin päätelaitteina Garminin Nüvi 750 -navigaattoreita, koska ne olivat syksyllä 2009 ainoita käytössämme olleita laitteita, joille oli mahdollista toteuttaa mediasisältöä tukevia opastuspisteitä ilman ohjelmointiosaamista. Myös Hattuvaaran syrjäinen sijainti ja esimerkiksi 3G-verkon puuttuminen rajoittivat päätelaitevaihtoehtoja.

GPS-Taipaleessa tarjottava sisältö oli palvelun luonteen takia tärkeässä asemassa. GPS-Taival oli elämyksellinen opastuspalvelu, jossa käyttäjälle tarjottiin

---

<sup>1</sup> Connected Mobile Communities in the Northern Periphery, <http://www.cmcnp.eu/>.

<sup>2</sup> GPS-Taival, <http://www.mobikarelia.fi>.

päätelaitteen välityksellä paikkaan sidonnainen rikastettu todellisuus. GPS-Taipaleessa todellisuutta rikastettiin kertomalla paikan historiaa. GPS-Taipaleen sisältönä olivat jokaiselle kolmelletoista opastuspisteelle toteutetut dramatisoidut ja näytellyt äänitiedostot, joiden avulla käyttäjä kuljetettiin pisteeltä toiselle tarinan muodossa. Äänikertomuksen lisäksi jokaisella opastuspisteellä oli yksi kuvatiedosto. Kuvatiedostot olivat pääasiassa taistelupaikalle jäänyttä esineistöä, jotka kuvattiin Taistelijan Talon sotahistoriallisessa museossa.

GPS-Taipaleessa käytettyjen navigaattoreiden tarkkuus aiheutti ongelmia opastuspisteiden asettelussa. Opastuspisteet sijaitsivat noin kilometrin mittaisella Taistelijan Taival -luontopolulla. Opastuspisteiden välimatkat vaihtelivat 100 metristä aina 10–15 metriin. Lähekkäin olevilla pisteillä paikkatiedon vaikutusalueiden koko jouduttiin pitämään mahdollisimman pienenä, että jokaisen pisteen vaikutusalue mahtuisi tiiviiseen opastusympäristöön. Päätelaitteen takia vaikutusalueiden muotona oli käytettävissä ainoastaan opastuspisteen ympärille muodostuva kehä. Tämän vuoksi vaikutusalueiden lomittaissijoittelu ei ollut mahdollista. Ongelmat saatiin kuitenkin ratkaistua sijoittamalla vaikutusalueiden keskipisteet opastuspisteiden viereen siten, että opastuspiste sijaitsee vaikutusalueella, mutta on huomattavasti sivussa sen keskipisteestä. Tämän ratkaisun myötä opastuspisteiden toimintavarmuus kuitenkin heikkeni huomattavasti. Tiheään sijoiteltujen opastuspisteiden yhteydessä oman haasteensa toi myös ympäröivä maasto, sillä se oli pääasiassa rinteessä sijaitsevaa tiheää kuusimetsää. Tämä aiheutti varsinkin kesällä ongelmia GPS-yhteydessä taivaalla oleviin satelliitteihin.

Autonavigointiin tarkoitetun laitteen soveltuvuus maastossa esitettävään mediasisällön toistamiseen on huono, sillä laitteen ominaisuuksia ei selvästikään ole suunniteltu tämän kaltaiseen käyttöön soveltuviksi. Valmiiseen laitevalmistajan tekemään toimintoon perustuvan opastuspalvelun sisällöt olivat rajoitettu yhteen opastuspistekohtaiseen äänitiedostoon ja yhteen kuvatiedostoon. Ongelmia ilmeni myös akunkestossa ja käytettävyydessä, sillä esimerkiksi 13 opastuspisteestä koostuneen ja noin 1,5 kilometriä pitkän Taistelijan Taipaleen kiertäminen oli navigaattorin akun suorituskyvyn ääri rajoilla. Käytettävyydeltään laite toimi automaattisena sijaintiin liitetyn äänimateriaalin toistimena hyvin, mutta sisällön löytäminen ja toistaminen monien valikoiden kautta oli hankalaa.

### 3.3 Turusen taival

Turusen taival<sup>1</sup> oli GPS-Taivalta seurannut, vastaavalla tekniikalla toteutettu projekti Lieksan Vuonisolahteen. Turusen taipaleessa kohteena olivat kirjailija Heikki Turusen lapsuusmaisemat Vuonislahden kylällä ja niihin liittyvät muistot kirjailijan itsensä kertomana. Turusen taival erosi sisällöiltään huomattavasti GPS-Taipaleesta, sillä siinä sisällön määrä oli huomattavasti pienempi eikä siinä ollut internetsisältöjä. GPS-Taipaleen käyttäjäkokemusten myötä havaitsimme kuva- ja tekstisisältöihin tutustumisen suurelle osalle käyttäjäkuntaa niin vaikeaksi, että Turusen taipaleessa tyydyimme tarjoamaan pelkkää äänisisältöä. Pelkkään äänisisältöön keskittyminen teki toteutuksesta enemmän laajennetun todellisuuden palvelun, koska päätelaitteen näkyvyys palvelun käytössä väheni.

Turusen taival toteutettiin veneillä Vuonislahden rantoja kierrettäväksi kertomukseksi. Palvelussa oli viisi opastuspistettä sijoitettuna eri puolille Pielisen Vuonislahden puoleista rantaa. Turusen taipaleessa emme kohdanneet aiemmin GPS-Taipaleessa ilmenneitä opastuspisteiden teknisiä ongelmia, sillä se oli toteutusympäristönsä ja pisteiden sijoittelun osalta huomattavasti helpompi kohde.

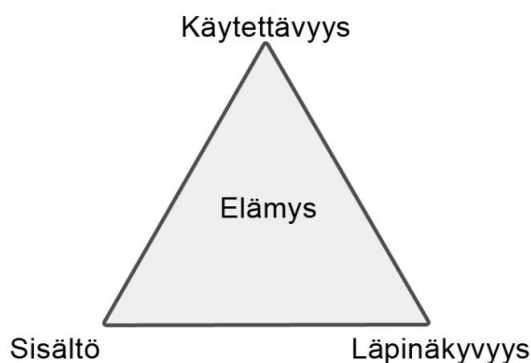
### 3.4 Palveluiden jatkokehitys

GPS-Taipaleessa ja Turusen taipaleessa suurimmat ongelmat olivat päätelaitteissa ja niiden käytettävyydessä. Nämä ongelmat vaikuttivat myös laajennetun todellisuuden elämykseen. Päätelaitteen huono käytettävyys teki mekanismeista sisältöön verrattuna liian keskeisen osan palvelua. Ongelmat poistuivat sisältöä vähentämällä. Tämä on kuitenkin ongelman ratkaisuksi kyseenalainen keino. Vaikka sisällön määrä tulee suhteuttaa tavoiteltavaan elämykseen ja siten rikastaa realismia mekanismien pysyessä piilotettuina, ei päätelaitteen huono käytettävyys saisi olla sisältöä rajoittavana tekijänä. Toteuttamissamme palveluissa sisällön määrä suhteutettiin enemmänkin päätelaitteen ominaisuuksien kuin palvelulla tavoiteltavan elämyksen mukaan. Pitäisinkin tasapainoilua käytettävyyden, tarjottavan sisällön ja mekanismien läpinäkyvyyden välillä keskei-

---

<sup>1</sup> Turusen taival, <http://www.mobikarelia.fi>.

simpänä laajennetun todellisuuden palveluita koskevana jatkokehityksen haasteena (ks. kuvio 1).



Kuvio 1. Laajennetun todellisuuden palveluiden elämyksellisyyteen vaikuttavat tekijät.

Saavutettujen kokemusten myötä virtuaaliopastuksen jatkokehityksen ensimmäisenä haasteena oli löytää päätelaite, jonka avulla esiintyneet ongelmat ovat kierrettävissä. Ohjelmointiympäristöjen ja kehittyneen laitekannan myötä laajennetun todellisuuden palvelut on mahdollista toteuttaa matkapuhelimille omina itsenäisinä sovelluksina (Kuivakari 2006, 8). Tämä mahdollistaa palvelun luonteen määrittelyn sisällön määrän ja halutun lopputuloksen mukaan. Matkapuhelinsovelluksina toteutetuissa laajennetun todellisuuden palveluissa tärkeimpiä tekijöitä ovat käyttöliittymä ja palvelun käytettävyys. Eri valmistajien matkapuhelimet ovat erilaisia käyttää, mutta koska opastussovellus on itsenäinen päätelaitteella suoritettava palvelu, on käytettävyys riippuvainen ainoastaan sovelluksen suunnittelussa tehdyistä ratkaisuista. Koska matkapuhelin on käyttäjälleen henkilökohtainen ja päivittäisessä käytössä oleva väline, on käyttäjä jo valmiiksi tietoinen sen käytettävyydestä. Oman puhelimen tuomalla lisäarvolla voidaan peittää myös pieniä sovelluksien käytettävyyso ongelmia sillä suoritetuissa sovelluksissa. (Ristolainen 2007, 17–18.) Itsenäisesti toimivan sovelluksen etuna – esimerkiksi navigaattoreihin verrattuna – on sen suunnittelu ja muokattavuus sisällön tuomien vaatimusten mukaisesti.

## **4 VUONISLAHDEN KYLÄKÄVELY**

### **4.1 Palvelun suunnittelu**

Itsenäisesti toimivan matkapuhelinsovelluksen pilotiksi olen valinnut palvelun nimeltä Vuonislahden kyläkävely. Palvelu toteutetaan syyskuun 2010 ja maaliskuun 2011 välisenä aikana yhteistyössä Pohjois-Karjalan ammattikorkeakoulun TG4NP-hankkeen kanssa. Palvelun kohteena on Lieksassa sijaitseva Vuonislahden kylä ja sen paikallishistoria. Vuonislahden kyläkävely -mobiilipalvelu toteutetaan aiemmista toteutuksista löytämieni puutteiden ja ongelmien pohjalta matkapuhelimella toimivaksi laajennetun todellisuuden opastuspalveluksi. Toteutan palvelusta opinnäytetyöni yhteydessä ainoastaan pilottiasteen version kahta eri pohjaratkaisuja käyttäen. Toteutetut pilottiversiot jäävät valmistuttuaan TG4NP-hankkeen käyttöön esittely- ja jatkokehitystarkoitukseen eikä niitä julkaista virallisesti.

### **4.2 Sovelluskehityksen lähtökohdat**

Aloitin opinnäytetyöni toiminnallisen osuuden suunnittelun kesällä 2010. Olin mukana Ilosaarirock-mobiiliopas-työryhmässä vastaamassa sen toteuttaman matkapuhelinsovelluksen teknisistä ratkaisuista. Ilosaarirock-mobiiliopas oli Ilosaarirock-festivaalille toteutettu matkapuhelimeen ladattava festivaaliopas, jonka tarkoituksena oli toimia vaihtoehtona jaettaville paperisille käsiohjelmille. Toteutimme sovelluksen JM-Mobile Editor -nimisellä Java ME -sovellusten toteutukseen tarkoitetulla ohjelmalla. Ilosaarirock-mobiiliopas oli minulle ensimmäinen koskaan toteuttamani matkapuhelinsovellus.

Sovelluksen toteutuksesta oppimieni asioiden ja lopputuloksesta saatujen hyvien palautteiden jälkeen aloin suunnitella opinnäytetyössäni toteutettavaa opastussovellusta. Aiemmissa palveluissa päätelaitteita koskevien ongelmien ja jatkokehitystarpeiden takia keskityin suunnittelemaan opastussovellusta ainoastaan matkapuhelinkäyttöön. Aiemmin toteutetuissa GPS-Taipaleessa ja Turusen taipaleessa ilmenneet päätelaiteongelmat, kuten akunkesto ja käyttöliittymä, oli mahdollista ratkaista käyttämällä matkapuhelimella suoritettavaa erillistä sovellusta.



Suunnittelin toteutusta ainoastaan Nokian matkapuhelimille. Nokian puhelinten valitsemiseen vaikuttivat aiemmin Ilosaarirock-mobiilioppaan toteutuksesta saadut myönteiset kokemukset. Ilosaarirock-mobiilioppaan toteutuksessa huomasimme, että JM-Mobile Editorilla toteutetut sovelluksen toimivat hyvin Nokian matkapuhelimeissa. Päätelaitevalintaan vaikuttivat myös laitteiden käyttökelpoisuus laajennetun todellisuuden palveluissa. Valitsin päätelaitteeksi Nokian kosketusnäyttölliset puhelinmallit, koska niissä on suuren näytön ansiosta riittävän iso sisällön esityspinta-ala sekä hyvä suorituskyky sisältöjen toistamiseen. Nokian valitsemiseen vaikuttivat myös käytännön syyt, sillä saatavillani oli kaksi erilaista Nokian kosketusnäyttöpuhelinta sovelluksen testaustarkoitukseen.

Aloittaessani sovelluksen suunnittelun oli JM-Mobile Editorilla toteutettava opastussovellus ainoa suunnittelemani vaihtoehto. JM-Mobile Editorilla tehtävä Java ME -sovellus vaikutti suunnittelun alussa hyvältä ratkaisulta, koska minulla oli siitä runsaasti taustatietoa ja myönteisiä kokemuksia. Seurasin sovelluksen suunnittelutyön aikana paljon Nokian sovelluskehittäjille tarkoitettua Forum Nokia<sup>1</sup> -yhteisöpalvelua. Forum Nokia -palveluun ilmaantui kesän ja syksyn 2010 aikana runsaasti keskusteluja ja tietoa WRT-ohjelmointiympäristöllä toteutetuista matkapuhelinsovelluksista. WRT:n tekniikka alkoi kiinnostaa syksyn 2010 aikana yhä enemmän, joten päätin opetella toteuttamaan vertailukelpoisen sovelluksen myös sitä käyttäen. Tavoitteenani oli omaksua WRT-tekniikka työkaluksi sovelluskehitykseen ja toteuttaa Java ME -toteutukseen vertailtava sovellus. WRT-tekniikan opetteluun yhteydessä lukemani keskustelut ja artikkelit Forum Nokia -palvelussa vakuuttivat paikkatietoa käyttävän opastuspalvelun toteutuksen olevan mahdollista myös WRT:llä.

### **4.3 Ohjelmointiympäristöt**

#### **4.3.1 Rajaus**

Sovelluksen toteutukseen matkapuhelimille on olemassa useita tapoja. Eri matkapuhelinvalmistajat tukevat lukuisia erilaisia ohjelmointiympäristöjä ja sovellusformaatteja. (Pihlajamäki 2010.) Keskityn työssäni kahteen hyvin erilaiseen oh-

---

<sup>1</sup> Forum.Nokia, <http://www.forum.nokia.com/>.

jelmointiympäristöön, jotka ovat Java-ohjelmointikielen kevytversio Java ME ja erilaisia verkkosivutekniikoita yhdistelevä Nokia WRT. Opinnäytetyön tavoitteena on selvittää näiden kahden ohjelmointiympäristön soveltuvuus laajennetun todellisuuden opastuspalvelun toteutukseen. Toteutan sovelluksesta pilottiversiot molempia toteutustapoja käyttämällä. Olen valinnut nämä kaksi ohjelmointiympäristöä, koska ne ovat toimintaperiaatteiltaan keskenään hyvin erilaisia. Toteutan sovelluksien pilottiversiot samoilla sisältömateriaaleilla ainoastaan sovellusympäristöä vaihtamalla.

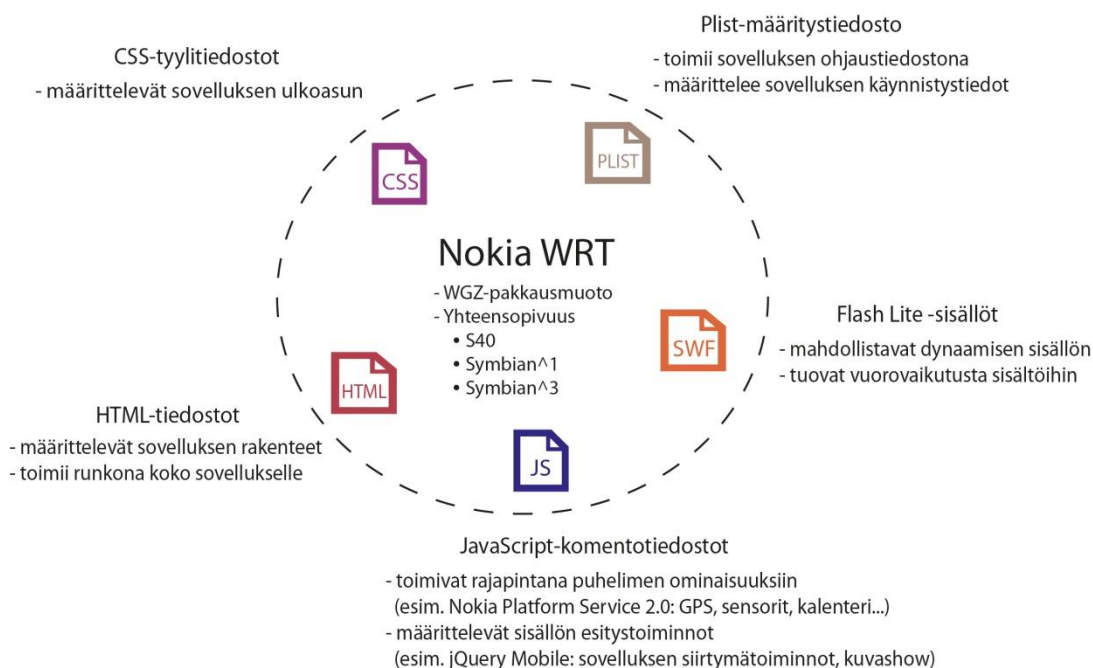
### **4.3.2 Java ME**

Java ME (Java Platform Micro Edition tai Java 2 Platform Micro Edition) on matkapuhelinkäyttöön optimoitu versio Java-ohjelmointiympäristöstä. Java ME on normaalia Javaa kevyempi suorittaa sen rajoitettujen ominaisuuksien ja luokkakirjastojen vuoksi. Java ME soveltuukin perinteistä Javaa paremmin suorituskyvyltään heikoille laitteille kuten matkapuhelimille. Java ME -sovellukset suoritetaan matkapuhelimessa erillisen toistimen eli virtuaalikoneen välityksellä. Virtuaalikone on matkapuhelimessa sovelluksen taustalla ajettava prosessi, jonka avulla toteutettu sovellus ja sen toiminnot tuodaan käyttäjälle näkyviin. Virtuaalikoneen takia ohjelmien suorittaminen vaatii kaksi päällekkäistä prosessia; taustalla ajettavan virtuaalikoneen ja itse sovelluksen. Java ME:n etuna on sen hyvä yhteensopivuus vanhojen puhelinmallien kanssa, vaikkakin suoritettavia toimintoja joudutaan rajoittamaan heikon suorituskyvyn takia. (Java User Group 2010.)

Käytin Java ME -sovelluksen toteutuksessa JM-Mobile Editor -ohjelmaa. Ohjelma on WYSIWYG-käyttöliittymällä toimiva visuaalinen editori matkapuhelinsovellusten toteutukseen. JM-Mobile Editorilla toteutetut sovellukset pystyvät käyttämään hyödyksi muutamia puhelimen sisäisiä ominaisuuksia. Sovelluksissa voidaan käyttää hyväksi tietoa puhelimen kellosta, GPS-sirun antamasta sijaintitiedosta sekä puhelimen sisäisistä liikkeentunnistussensoreista. Editorin käyttö ei vaadi suurta ohjelmointitaitoa, sillä se tekee ohjelmoinnin käyttäjän asettamien visuaalisten elementtien mukaan. Ainoastaan toteutetussa sovelluksessa käytettävien muuttujien ohjelmointi jää käyttäjän tehtäväksi. (JM-Mobile Solutions 2008.)

### 4.3.3 WRT

WRT (Web Runtime) ei itsessään ole ohjelmointiympäristö, vaan se on yhdistelmä erilaisia web-suunnittelussa käytettyjä ohjelmointikieliä. WRT toimii jakelumuotona sovellukselle, jossa voidaan yhdistää HTML-, CSS-, JavaScript- ja Flash Lite -tiedostoja (ks. kuvio 2). WRT:llä toteutettujen sovellusten rakenne perustuu perinteisten verkkosivujen rakenteisiin. WRT-tekniikan avulla matkapuhelimessa pystytään suorittamaan rakenteeltaan verkkosivumaisia sovelluksia paikallisesti ilman verkkoyhteyttä. WRT-lyhennelmää käytetään Nokian matkapuhelimille toteutetuissa sovelluksissa, mutta vastaavalla periaatteella voidaan toteuttaa sovelluksia myös muiden puhelinvalmistajien matkapuhelimille ainoastaan tiedostorakenteita muuttamalla. Muiden matkapuhelinvalmistajien vastaavaan toteutustapaan perustuvia sovelluksia kutsutaan usein verkkosovelluksiksi (Web Application). WRT-tekniikalla toteutetut sovellukset suoritetaan suoraan puhelimessa ilman virtuaalikonetta. (Nokia 2010.)



Kuvio 2. WRT:llä toteutetun sovelluksen rakenne.

WRT:ssä sovellukset suoritetaan suoraan puhelimessa ilman virtuaalikonetta. WRT-tekniikassa toiminta jakautuu useiden eri komentokielen välille: HTML-komentokielellä määritellään sovelluksen rakenne, CSS:llä visuaalinen ulkoasu ja asettelu, JavaScript-komentokielellä sovellukseen voidaan yhdistää erilaisia vuorovaikutteisia toimintoja ja päätelaitteen ominaisuuksia ja Flash Lite

-sisältöjen avulla sovellukseen voidaan tuoda dynaamisia sisältöjä, kuten esimerkiksi animaatioita tai liikkuvia elementtejä. Koska WRT:ssä useat eri komentokielet määrittelevät omat osuutensa sovelluskokonaisuudesta, on sovelluksen päivittäminen ja sisällönhallinta yksinkertaista. Esimerkiksi pelkästään ulkoasua päivitettäessä muutostyöt tehdään ainoastaan sovelluksen CSS-tiedostoihin ilman että sovelluksen muuta rakennetta tarvitsee muuttaa. WRT-sovelluksen sisällönhallinta onkin hyvin verrattavissa moderniin internetsivujen sisällönhallintaan. (Blandford & Spence 2008.)

Päätelaitteen ominaisuuksiin yhdistäminen tapahtuu WRT:ssä erillistä JavaScript-rajapintaa käyttämällä. Yksi vaihtoehto Nokian puhelimille toteutettaessa on Nokian kehittänyt Nokia Platform Services 2.0 -rajapinta. Platform Services 2.0 -rajapinnan kautta WRT-sovelluksella on mahdollisuus saada tietoa esimerkiksi puhelimen GPS-sirun tarjoamasta paikkatiedosta, puhelimen asentoa seuraavista kiihtyvyyssensoreista ja esimerkiksi käyttäjän tekemistä kalenterimerkinnöistä. Täydellinen listaus käytettävistä ominaisuuksista on nähtävissä Nokian Platform Services -kehittäjäkirjastossa<sup>1</sup>. (Nokia 2010.)

WRT-sovellukset voidaan toteuttaa millä tahansa ohjelmointikielten kirjoitukseen sopivalla ohjelmalla. Käytin itse minulle verkkosivutoteutuksista entuudestaan tuttua Adobe Dreamweaver CS5 -ohjelmaa. Valitsin Dreamweaverin mobiilipalvelun toteutukseen, koska sillä työskentely on jo entuudestaan tuttua ja siihen on saatavilla hyviä laajennuspaketteja mobiiliohjelmointiin. Käytin toteutuksessa apuna Nokian tarjoamaa WRT-laajennuspakettia<sup>2</sup>. Laajennus tarjoaa Dreamweaveriin hyvät lisätoiminnot WRT-sovellusten hallintaan ja valmiin sovelluksen koostamiseen.

Koska WRT-tekniikassa käytetään samoja komentokieliä kuin verkkosivuissa, ovat WRT-pohjaiset sovellukset helposti muunnettavissa myös verkossa julkaitavaksi. WRT-tekniikkaa käyttämällä voidaan yhdistää erilliset verkkosivut ja sovellus yhdeksi internetissä suoritettavaksi verkkosovellukseksi. Samaa verkkosovellusta voidaan käyttää erilaisilla laitteilla, kuten esimerkiksi tietokoneilla ja matkapuhelimilla. WRT-tekniikan ja rajapintojen avulla verkkosovellukseen pys-

---

<sup>1</sup> Nokia Web Developer's Library 2.0. Forum.Nokia 2010. <http://library.forum.nokia.com/>.

<sup>2</sup> Nokia WRT Extension for Adobe Dreamweaver. Forum.Nokia 2010. [http://www.forum.nokia.com/Develop/Web/Tools/Dreamweaver\\_extension/](http://www.forum.nokia.com/Develop/Web/Tools/Dreamweaver_extension/).

tytään tuomaan tietoa päätelaitteesta. Matkapuhelin voi ilmoittaa verkkosovellukselle esimerkiksi käyttäjän sijainnin tai tietoa hänen käyttämistä yhteisöpalveluista. Tämän avulla verkkosivut pystytään toteuttamaan yhä enemmän vuorovaikutteiseksi käyttäjän kanssa. (Blandford & Spence 2008.)

## **4.4 Toteutuksen suunnittelu ja esivalmistelut**

### **4.4.1 Tavoite**

Opinnäytetyöni toiminnallisena tavoitteena oli toteuttaa kahdella eri tekniikalla toimivat pilottiversiot Vuonislahden kyläkävely -matkapuhelinsovelluksesta. Vaikka kyseessä oli ainoastaan pilottiversioiden toteutus, oli tavoitteenani tehdä niistä teknisesti mahdollisimman viimeistellyt. Aloitin toteutuksen suunnittelun syyskuussa 2010. Suunnittelun tuloksena toteutin rakennekaavion kuvaamaan sovelluksien perusrakennetta ja sisällön asettelua (ks. liite 1). Kiinnitin erityisesti huomiota käyttöliittymäsuunnitteluun, sillä aiempien toteutusten kehitystarpeet olivat juuri päätelaitteen käyttöliittymässä ja toiminnan yksinkertaistamisessa.

### **4.4.2 Käyttöliittymäsuunnittelu**

Käyttöliittymällä (user interface) tarkoitetaan sovelluksen sisällön ja käyttäjän välisen vuorovaikutteisuuden mahdollistavaa toimintokokonaisuutta. Käyttöliittymä on käyttäjän näkemä, kuulema tai tuntema osa sovellusta, jonka välityksellä käyttäjä käyttää sovelluksen tarjoamia palveluita ja selaa sovelluksen sisältöä (IBM 2011). Aloitin Vuonislahden kyläkävely -sovelluksen käyttöliittymäsuunnittelun samaan aikaan rakenteellisen suunnittelun kanssa syyskuussa 2010.

Koska olin valinnut päätelaitteeksi kosketusnäytölliset puhelimet, tuli käyttöliittymäsuunnittelussa ottaa huomioon kosketuskäytön asettamia vaatimuksia. Yksi kosketuskäyttöliittymän keskeisiä asioita on navigointipainikkeiden sijoittelu, koko ja kosketuspinta-ala. Käyttöliittymää ja rakennetta suunnitellessa käytin hyväksi myös aiemmista toteutuksista tekemiäni huomioita navigaattoreiden käytettävyydestä ja niiden käyttöliittymien tarpeettomasta monimutkaisuudesta.

Tavoitteena oli toteuttaa rakenteeltaan mahdollisimman yksinkertainen ja helpotoiminen käyttöliittymä. Tavoitteenani oli tehdä käyttöliittymästä helppokäyttöinen myös sellaisille henkilöille, jotka eivät ole kosketusnäyttöistä puhelinta ennen käyttäneet. Lähtökohtina helppokäyttöisyyteen oli sovelluksen ylimääräisten valikkojen vähentäminen ja sisällön suora esittäminen omien alisivujen yhteydessä. Myös riittävän isot navigointipainikkeet tekevät sovelluksen käytöstä helpompaa varsinkin henkilöille, joille kosketuspuhelin käyttö on entuudestaan tuntematonta. Riittävän isoilla navigointipainikkeilla turhauttavat ohjainalukset vähenevät ja sovelluksen käyttö on helpompaa myös isosormisille käyttäjille. (Waloszek 2000.)

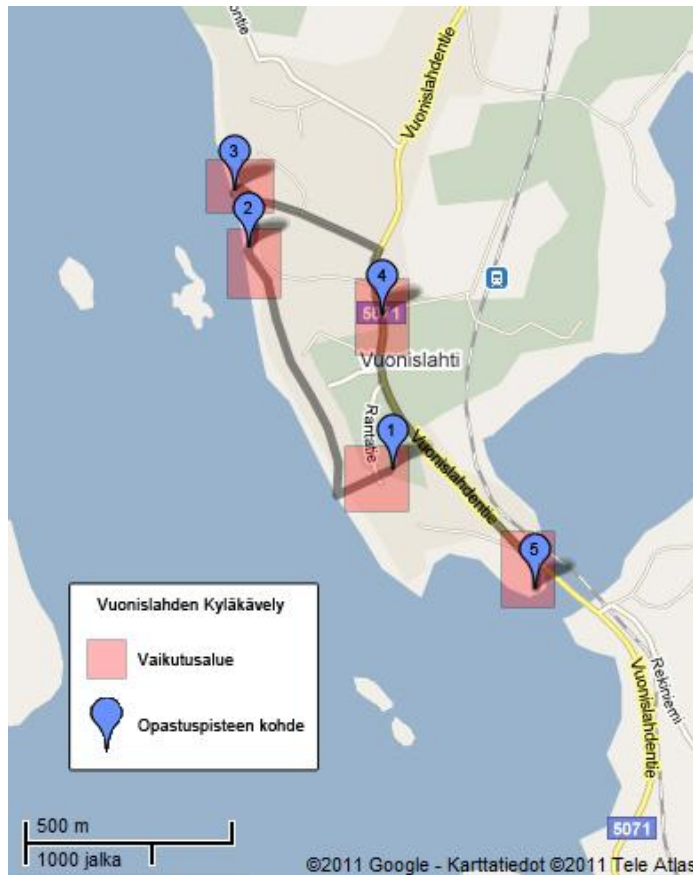
Tekemäni ratkaisu yksinkertaisesta sovelluksen sisältörakenteesta osoittautui käyttöliittymää suunniteltaessa hyväksi. Koska sovelluksen sisällöt oli jaettu etusivujen ja erillisten sisältösivujen alle, jäi yksittäisten sivujen sisältömäärä varsin kohtuulliseksi. Tämä helpotti käyttöliittymäsuunnittelua, koska vähäisten sisältömäärien vuoksi esimerkiksi navigointipainikkeiden sijoitteluun jäi reilusti näyttöpinta-alaa. Sormikosketuksellisten navigointipainikkeiden suosituspinta-ala on 2,2 cm<sup>2</sup> ja välistys 3 mm (Waloszek 2000). Toteutetuissa sovelluksissa pienimmät painikkeet olivat kosketuspinta-alaltaan 2,5 cm<sup>2</sup> ja välistys 5 mm.

Koska kyseessä on laajennetun todellisuuden mobiilipalvelu, ovat toimintojen helppokäyttöisyys ja automatisointi keskeisiä ominaisuuksia sisällön esiintuomisessa. Käyttöliittymäsuunnittelussa käytettävyyden on huomioitu tarjoamalla käyttäjälle erillinen etusivunäkymä ja erilliset automaattisesti käynnistyvät sisältösivut. Sovellusta käyttäessään käyttäjän ei tarvitse osallistua toimintojen käynnistämiseen muuten kuin fyysisesti liikkumalla opastuspisteiden sijaintiin. Halutessaan käyttäjä voi käynnistää opastuspisteet myös manuaalisesti ohjelman päänäkymästä. Sijoitin opastuspisteiden käynnistämisen ohjelman päänäkymään, sillä halusin sisältöjen selauksen olevan yksinkertaista ja mahdollisimman väliportaatonta. Sisältösivuilla laajennetun todellisuuden vaikutelma on huomioitu sisältöjen automaattisena käynnistymisenä, joten äänet ja kuvasisältö käynnistyvät automaattisesti käyttäjän saapuessa opastuspisteelle. Käyttäjä voi halutessaan pitää päätelaitetta taskussa ja keskittyä ainoastaan kuulokkeiden välityksellä automaattisesti toistuvaan pistekohtaiseen äänisisältöön.

#### 4.4.3 GPS-mittaukset

Ennen sovelluksen ohjelmointityön aloittamista suoritimme opastuspisteiden GPS-mittaukset Lieksan Vuonislahdessa. Mittaukset suoritettiin Tomtom GO 750 -navigaattorin ja Nokian N97 -matkapuhelimen antamaa GPS-paikkatietoa käyttäen. Kahden eri laitteen antamaa koordinaattitietoa vertailemalla saimme minimoitua virhemarginaalin ja siten tarkat tulokset heti ensimmäisellä mittauksella. Molemmat mittauksessa käytetyt laitteet oli asetettu antamaan koordinaatit desimaaliasteina.

Mittauksissa jokaisesta opastuspisteestä otettiin kolme arvoa: koillinen raja-arvo, opastuspisteen sijainti ja lounainen raja-arvo. Kolmen arvon mittauksella jokaisesta opastuspisteestä saatiin määritettyä vaikutusalueen raja-arvot sekä opastuspisteen sijainti. Valitsin mittaukseen koillisen ja lounaisen arvon, koska niiden avulla mittaustuloksista saadaan desimaalilukuina alueen nurkkaukset määrittävät arvot. Koillinen raja-arvo pituus- ja leveyspiirien desimaaliasteina on arvoltaan alueen suurin koordinaattiluku ja lounainen raja-arvo pienin. Opastuspisteen keskipiste mitattiin havainnollistamaan opastuspisteen sijoittumista mitattuun vaikutusalueeseen. Mittausten jälkeen syötin tulokset Google Maps -karttaohjelmaan ja määritin neliskulmaiset kuviot esittämään mitattuja vaikutusalueita. Vaikutusalueiden havainnollistaminen karttakuvan avulla helpotti huomioimaan mahdollisia ongelmakohtia, kuten päällekkäisiä vaikutusalueita, jo ennen varsinaisen ohjelmointityön aloittamista (ks. kuvio 3).



Kuvio 3. Opastuspisteiden vaikutusalueet kartalle havainnollistettuna.

#### 4.4.4 Sisältömateriaalit

Vuonislahden kyläkävely -matkapuhelinpalvelussa käytetään sisältömateriaalina ääni- ja kuvatiedostoja. Jokainen opastuspiste sisältää yhden äänitiedoston sekä 3–5 kuvatiedostoa. Sisältömateriaalit on toteutettu yhteistyössä Pohjois-Karjalan ammattikorkeakoulun TG4NP-hankkeen ja Vuonislahden Kyläyhdistyksen kanssa. Sovelluksen sisältömateriaalien tuotannosta vastasivat Pohjois-Karjalan ammattikorkeakoulun viestinnän koulutusohjelman opiskelijat.

Sisältömateriaalien tiedostokoot on optimoitu matkapuhelinkäyttöä varten. Äänien tiedostokoot ovat 500–900 kilotavua ja yhden äänitiedoston pituus on keskimäärin noin minuutti. Äänitiedostojen bittinopeus on laskettu 96 kilobittiin sekunnissa pienen tiedostokoon saavuttamiseksi. Kuvatiedostot ovat kooltaan 20–60 kilotavua, ja niiden kuvanlaatu on optimoitu matkapuhelimen näytöltä katseltavaksi.



## **4.5 Java ME -sovellusversio**

### **4.5.1 JM-Mobile Editor**

Aloitin ensimmäiseksi Java ME -sovellusversion tekemisen, koska sen toteutamisessa työkaluna käyttämäni ohjelmisto JM-Mobile Editor oli minulle entuudestaan tuttu Ilosaarirock-mobiilioppaan toteutuksesta. Automaattista ohjelmointityötä tekevien editoreiden työn jälki on usein tarpeettoman monimutkaista ja siten myös suorituskyvyltään heikompaa (Carboni ym. 2011). JM-Mobile Editorissakaan Java-ohjelmointityö ei vastaa ominaisuuksiltaan ja suorituskyvyltään täysin käsin kirjoitettua Java-koodia, mutta on pienistä puutteista huolimatta mielestäni vertailukelpoinen kevyttä ohjelmointiosaamista vaativan WRT:n kanssa. JM-Mobile Editorin tekemä automatisoitu ohjelmointityö on kuitenkin Java ME -kieleksi hyvin suorituskykyistä ja rakenteeltaan kehittyntä, ja se soveltuu hyvin kevyttä ohjelmointia vaativien matkapuhelinsovellusten toteutukseen. Esimerkkinä JM-Mobile Editorin tekemän ohjelmoinnin laadusta on vuoden 2010 Ilosaarirock-mobiiliopas, joka oli automatisoidusta ohjelmointityöstä huolimatta suorituskykyinen ja toimintavarma matkapuhelinpalvelu. Aloittaessani sovelluksen toteutusprosessin oli käytössäni ohjelman versio 1.3.26, mutta prosessin aikana se päivittyi 1.3.29-versioon. Käyttämäni JM-Mobile Editorin versioilla ei ollut vaikutusta lopputulokseen, vaan tarjotut päivitykset vaikuttivat ainoastaan ohjelman vakauteen ja suorituskykyyn vanhemmilla tietokoneilla käytettäessä. (JM-Mobile Solutions 2008.)

### **4.5.2 Sovelluspohjan määrittely**

Aloitin Java ME -sovellusversion tekemisen sovelluksen pohjarakenteiden toteutuksella ja ominaisuuksien määrittelyllä. JM-Mobile Editorissa sovelluksen rakenne koostuu erillisistä näkymistä (scene), jotka toimivat joko toiminta-aikaan sidottuina näytöksinä tai omina erillisinä osioina. Toiminta-aikaan perustuvalla toteutustavalla sovelluksesta saadaan tehtyä rakenteeltaan enemmän esitysmäinen, kun taas erillisillä osioilla toteutettu sovellus on rakenteeltaan enemmän käyttäjän tekemiin valintoihin ja navigointiin perustuva. Käytin Vuo-

nislahden kyläkävely -sovelluksessa käyttäjän navigointiin perustuvia osioita, koska halusin sovelluksen olevan vuorovaikutteinen käyttäjän kanssa.

Ennen sovelluksen rakenteiden toteutusta määrittelin sovellukselle muutamia sen esitystapaa koskevia ominaisuuksia. JM-Mobile Editorissa pystytään toteuttamaan näyttökooltaan käytännössä minkä kokoisia sovelluksia tahansa. Kesälä 2010 toteutetun Ilosaarirock-mobiilioppaan prosessin yhteydessä huomioin, että JM-Mobile Editorilla toteutetut sovellukset ovat todella huonoja skaalautumaan käyttäjän puhelimen mukaisesti. Huonon skaalautuvuuden vuoksi onkin tärkeää määritellä sovelluksen lopullinen näyttökoko JM-Mobile Editorin asetuksista ennen toteutusprosessin varsinaista aloittamista. Käytin kyläkävely-palvelussa Nokian kosketusnäyttöpuhelinin vaakasuuntaista esityspinta-alaa (640 x 360 px) sovelluksen näyttökokona.

Sovelluksen ominaisuuksien määrittelyn jälkeen toteutin sovelluksen rungon tyhjillä näkymillä. Tyhjän rungon avulla pystyin helpommin hahmottamaan sovelluksen koko rakenteen JM-Mobile Editorissa. Rungon tekeminen helpotti myös koostamistyötä, koska siten sisällöt on helpompi asetella jo olemassa oleville osioille. Alkuperäinen suunnitelmani oli toteuttaa sovelluksen runko siten, että jokaisella opastuspisteellä on oma erillinen näkymänsä (ks. kuvio 4). Lopullisessa sovelluksessa käytetyn verkkorakenteen vastaava runko on esitetty kuviossa 7.



Kuvio 4. Kuvakaappaus JM-Mobile Editorista. Kuvassa on sovelluksen rakenteen runko tyhjiillä osioilla (scene) toteutettuna.

#### 4.5.3 Verkkopohjainen XML-rakenne

Java-versiossa ilmeni suorituskykyongelmia isojen sisältömiärien esittämisessä. Ensimmäisten sovelluksen koeversioiden jälkeen huomasin ohjelmointiympäristön sopivan huonosti raskaiden sisältöjen kuten ääni- ja kuvatiedostojen toistamiseen. Koska Java ME -sovelluksissa kaikki sisältö toistuu saman virtuaalikoneen välityksellä, vaikuttavat yksittäiset raskaat sisällöt koko ohjelman toimintaan. Tämän vuoksi ääni- ja kuvatiedostojen latautuminen yhtäaikaaisesti sovelluksen muun sisällön kanssa oli käytettävyyden kannalta liian hidasta.

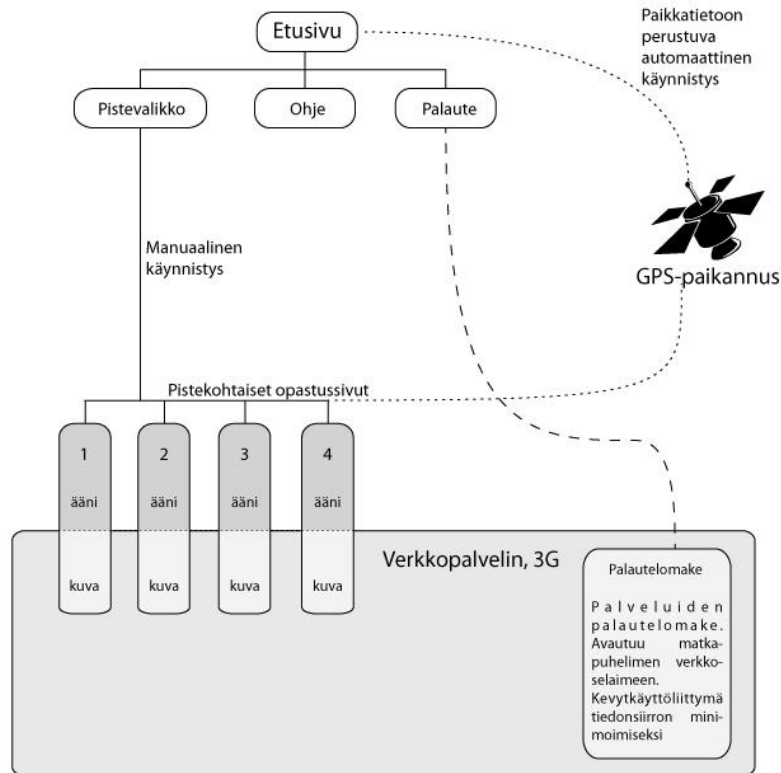
Raskas sisältömateriaali hidasti koko sovelluksen suorituskykyä, ja esimerkiksi navigointipainikkeiden toimintaviiveet olivat liian suuria. Ratkaisin latausongelmat siirtämällä raskaat sisältömateriaalit verkkopalvelimelle, josta sovellus voi ne tarvittaessa 3G-yhteydellä hakea. Tämä paransi sovelluksen käynnistymisaikaa ja osioiden välisiä siirtymäviiveitä, mutta ratkaisun takia sovellus toimii ainoastaan 3G-verkon peittoalueilla. Opastuksen kohdealueella 3G-yhteys toimii kuitenkin hyvin.

Koska 3G-yhteyden käyttö oli raskaiden sisältötiedostojen vuoksi välttämätöntä, otin sen huomioon sovelluksen muissakin ominaisuuksissa. Toteutin sovelluksen rakenteen verkkopohjaisesti XML-taulukon avulla. Käytännössä sovellus hakee jokaisella käynnistyskerralla tiedot valikkorakenteista ja sisältöjen määrästä erillisestä verkkolevyllä sijaitsevasta XML-tiedostosta. XML-tiedostossa on luettelomaisesti ilmaistu sovelluksen erilliset sisältösivut sekä osoitettu niihin ladattavat kuva- ja äänitiedostot (ks. kuvio 5). Määrittelin JM-Mobile Editorissa sovelluksen sisällöt siten, että jokaisen opastuspisteen tiedostonimet haetaan XML-taulukosta. Kun käyttäjä valitsee päävalikosta esimerkiksi opastuspisteen numero yksi, tallentuu valitun pisteen numerotieto sovelluksen välimuistiin. Sovellus vertaa välimuistissa olevaa pistenumeroa XML-taulukkoon ja näyttää pistenumeroa vastaavat sisällöt käyttäjälle. Sisällön hakeminen XML-taulukosta sovellukseen on esitetty graafisesti liitteessä 2.

```
<?xml version="1.0" encoding="utf-8"?>
<maincontent>
  <item id="1"> <!-- OPASTUSPISTEEN NUMERO -->
    <title>Pistel</title> <!-- OPASTUSPISTEEN NIMI, NÄKY YMYÖS ETUSIVUN VALIKOSSA -->
    <valokuvat>
      <kuva id="1"> <!-- KUVAN NUMERO -->
        <kuvatiedosto>pistel_kuval.jpg</kuvatiedosto> <!-- KUVAN TIEDOSTONIMI -->
      </kuva>
      <kuvamaara>4</kuvamaara> <!-- OPASTUSPISTEEN KUVIEN LUUKUMÄÄRÄ -->
    </valokuvat>
    <aaniopastus>
      <aanitiedosto>Pistel.mp3</aanitiedosto> <!-- ÄÄNEN TIEDOSTONIMI PALVELIMELLA -->
    </aaniopastus>
  </item>
</maincontent>
```

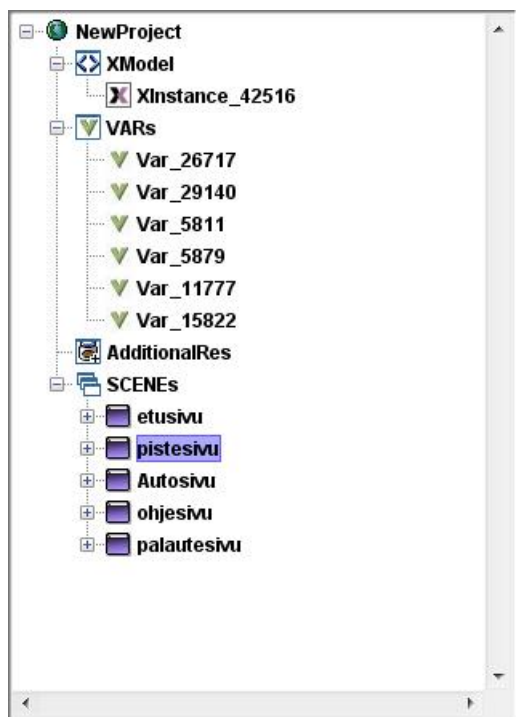
Kuvio 5. Esimerkki sovelluksen käyttämästä XML-rakenteesta.

Sovellukseen kehittämäni rakenteen (ks. kuvio 6) kautta sen sisältö ja sisältösivujen määrä on suoraan muokattavissa ilman sovelluksen ohjelmointikoodin muokkaamista. Tämän ansiosta esimerkiksi sisällöntarjoaja voi muokata sovelluksessa esitettävää sisältöä ilman ohjelmointikielien tuntemusta. XML-taulukkoa käyttävän rakenteen ansiosta sovelluksen erilliset opastuspistesivut voidaan yhdistää yhden sisältösivun alaisuuteen ja ainoastaan sisällön elementit (kuvat, ääni) ovat pistenumeron mukaan vaihtuvia.



Kuvio 6. Java ME -sovellusversion verkkorakenteiden toimintamalli.

Kuten kuvio 6 ilmenee, jouduin jättämään kaikkein raskaimmat sisällöt eli äänitiedostot sovelluksen sisältä ladattavaksi. Tämä johtui käyttämästäni JM-Mobile Editor -ohjelmasta, sillä se ei tue äänitiedostojen lataamista verkkoyhteyden välityksellä. Sisältöjen toistamista on käsitelty opinnäytetyön luvussa 4.6.3.



Kuvio 7. Kuvakaappaus JM-Mobile Editorista. Kuvassa on XML-taulukkoon perustuvan sovellusrakenteen runko tyhjiillä osioilla (scene) toteutettuna.

#### 4.5.4 Sisältömateriaalien toistaminen

JM-Mobile Editorilla toteutetuissa sovelluksissa on mahdollista toistaa ääntä, videokuvaa ja kuvatiedostoja. Ohjelma tarjoaa sisältömateriaalien toistamiseen muutamia yksinkertaisia aputoimintoja kuten diashow-toiminnon kuvatiedostoille, äänitoistimen äänitiedostoille sekä videotoistimen liikkuvalla kuvalla. Kaikki sisältömuodot toistetaan toteutetussa sovelluksessa ilman esimerkiksi puhelimen omien ääni- tai kuvatoistimien käyttöä. JM-Mobile Editor tukee äänen tiedostomuotona WAW- ja MP3-formaatteja. Kuvatiedostoja sovelluksessa voidaan esittää käytännössä kaikilla yleisimmillä kuvatiedostoformaateilla. Liikkuvan kuvan toistossa ohjelma tukee ainoastaan matkapuhelimissa paljon käytettyä 3GP-formaattia.

Käytin sovelluksessa kuvatiedostoissa JPEG-tiedostomuotoa ja äänitiedostoissa tiiviisti pakattua MP3-tiedostomuotoa. Vaikka sisältömateriaalit oli optimoitu tiedostokokojensa puolesta matkapuhelinkäyttöön soveltuviksi, oli sisältöjen määrä liian suuri Java ME -sovelluksen suorituskyvylle. Raskaiden sisältöjen takia sovelluksen latausajat pidentyivät ja käytöstä tuli liian hidasta. Suorituskyvyn parantamiseksi jouduin siirtämään kuvasisällöt 3G-verkon välityksellä

ladattavaksi. Kuvatiedostojen optimoinnin ansiosta kuvien tiedostokoot olivat 20–60 kilotavua, joten niiden lataaminen 3G-verkon yli oli sujuvaa. Halusin esittää kuvasisällöt siten, että vain yksi kuva näkyisi kerrallaan mahdollisimman suurikokoisena käyttäjälle. Toteutin JM-Mobile Editorin avulla sovellukseen kuvaesityksen, jossa yksittäisen kuvan esityspinta-ala oli 455 x 255 pikseliä. Kuvatoistin hakee opastuspisteen kuvien lukumäärän ja kuvien tiedostonimet XML-taulukosta (ks. liite 2).

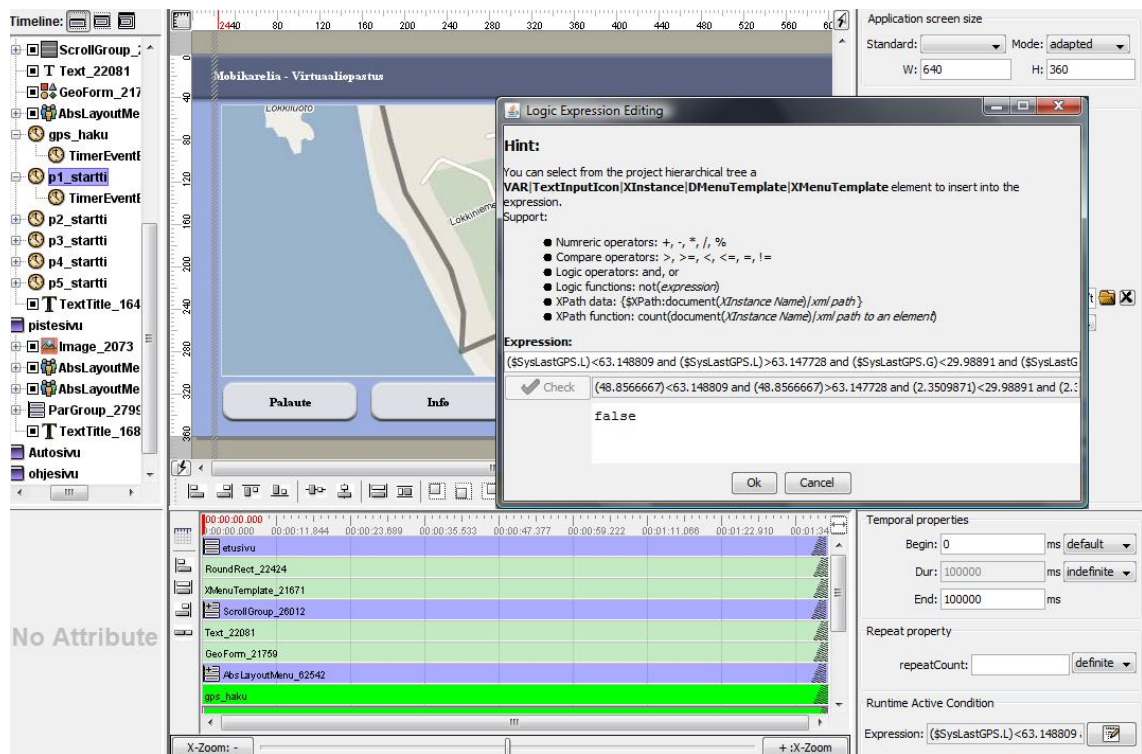
Käytin äänitiedostoissa tiiviisti pakattua MP3-formaattia. Äänitiedostojen toistaminen JM-Mobile Editorilla toteutetuissa sovelluksissa on mahdollista ainoastaan sovelluksen sisäisistä tiedostoista. JM-Mobile Editorin käyttämä äänitoistin tarvitsee äänitiedon aikatiedot eli äänen aloitusajan ja lopetusajan. Tietojen hakeminen ei ole mahdollista verkkopalvelimella sijaitsevista äänitiedostoista. Tämän takia jouduin pitämään äänitiedostot sovelluksen sisään liitettyinä. Toteutin äänisisällöt XML-taulukosta haettavilla tiedostonimillä, koska olin käyttänyt vastaavaa rakennetta jo kuvasisällöissä.

#### **4.5.5 Paikkatiedon yhdistäminen sisältöihin**

JM-Mobile Editor mahdollistaa muutamien puhelimen ominaisuuksien hyötykäytön toteutetuissa sovelluksissa. Puhelimesta voidaan tuoda sen kellonaika ja GPS-sirun tarjoama sijainnin koordinaatit. Käytin Vuonislahden kyläkävely-sovelluksessa puhelimen tarjoamaa sijaintitietoa yhdistämään tarjottavat sisällöt opastuspisteiden sijainteihin.

Toteutin GPS-tiedon yhdistämisen JM-Mobile Editorissa sen tarjoamien valmiiden sijaintitiedon hakutoimintojen avulla. Toteutettavaan sovellukseen määritellään aikaan sidonnainen paikkatietoa hakeva elementti (Time Interval Element), jolle voidaan määritellä haluttu toiminto. Elementille asetetaan haluttu toimintoväli, eli miten usein elementti suorittaa toiminnon, ja haluttu toiminto, joka tässä tapauksessa on GPS-informaation haku. Toteutin elementin siten, että se hakee ja päivittää puhelimen sijaintitiedon yhden sekunnin välein. Mitä tiheämmin sovellus hakee GPS-informaatiota, sitä enemmän se käyttää puhelimen akkua. Tällä päivitysnopeudella sovellus reagoi käyttäjän liikkeisiin nopeasti, mutta säästää kuitenkin akkua hakemalla paikkatiedon ainoastaan joka toinen sekunti.

Paikkatieto haetaan sovellukseen JM-Mobile Editorin valmiin laitemuuttaja-toiminnon avulla (System Variable). Paikkatietoon yhdistettävälle sisällölle asetetaan mitatut raja-arvot desimaalimuotoisina koordinaatteina JM-Mobile Editorin loogisiin ehtolausekkeisiin (logic expressions). Näitä raja-arvoja vertaillaan puhelimen sijaintitietoihin, ja kun puhelimen sijainnista kertovat koordinaatit ovat raja-arvojen sisällä, käynnistyy yhdistetty sisältö käyttäjälle näkyväksi.



Kuvio 8. GPS-paikkatiedon yhdistäminen sovelluksen sisältöön JM-Mobile Editorissa.

Kuviossa 8 on havainnollistettu JM-Mobile Editorin logic expression editing -toiminto, jonka avulla asetetaan opastuspisteiden koordinaattien raja-arvot ja niiden vertailukomennot puhelimen tarjoaman GPS-informaation kanssa. Toteutettiin puhelimen sijaintitietoja seuraavat elementit sovelluksen etusivulle, koska mielestäni etusivu on loogisin käyttäjän säilyttämä näkymä pisteeltä toiselle siirryttäessä. Tein jokaiselle opastuspisteelle oman käynnistys-elementin, joka seuraa puhelimen sijaintitietoja ja käynnistää opastuspisteen sisällön sijaintitietojen ollessa opastuspisteen raja-arvojen sisällä. Sijaintitietojen seuranta toteutetaan JM-Mobile Editorissa loogisten lausekkeiden (logic operators) avulla. Loogisissa lausekkeissa pystytään asettamaan käytännössä rajaton määrä ehtolausekkeitä. JM-Mobile tukee ehtolausekkeissa and- ja or-ehdoja sekä suurempi kuin-, pienempi kuin- ja yhtä suuri kuin -relaatioita. Sijaintitietojen vertailussa käytin



lausekkeita, jotka vertailevat puhelimen tarjoamaa paikkatietoa. Kun paikkatiedon arvo on esimerkiksi pienempi kuin opastuspisteen sijainnin koillinen raja-arvo ja suurempi kuin lounainen raja-arvo, käynnistyy opastuspisteen materiaalisivu käyttäjälle näkyväksi.

#### **4.5.6 Sovelluksen ulkopuoliset sisällöt**

Toteutin sovellukseen erillisen käyttäjäpalautte-osion, jossa tavoitteeni oli testata ulkoisen internetsisällön yhdistämistä sovellusrakenteeseen. Käyttäjäpalautte toteutettiin ulkoisella linkityksellä erilliseen verkkosivuilla olevaan käyttäjäpalauttelomakkeeseen. Käytin sovellusversioissa Ilosaarirock-mobiiliopas-projektin yhteydessä toteutettua käyttäjäpalauttelomaketta. Käyttäjäpalauttelomake on toteutettu Joomla!-julkaisujärjestelmää käyttämällä. Käyttäjäpalautteeseen ohjaaminen on toteutettu normaaliin navigointipainikkeeseen JM-Mobile Editorin ulkoiseen linkitykseen tarkoitetulla hyperlink-toiminnolla. Ennen käyttäjäpalauttelomakkeeseen siirtymistä käyttäjälle kerrotaan lomakkeen sijaitsevan ulkoisella verkkosivulla. Käyttäjäpalauttelomakkeeseen siirtyessä sovellus avaa puhelimen oman verkkoselaimen sovellusikkunan päälle, joten palauttelomakkeessa vierailun jälkeen käyttäjä palaa takaisin kyläkävely-sovellukseen. Palauttelomakkeen kautta annettu informaatio tallentuu verkossa sijaitsevaan tietokantaan, josta se on sivuston hallintatyökaluilla nähtävissä.

#### **4.5.7 Sovelluksen julkaisu**

Java ME -sovellusversion jakelu toteutetaan JAR- ja JAD-tiedostomuotoja käyttäen. Sovellus pakataan JAR-pakkausformaattiin, joka sisältää käytännössä kaiken sovelluksessa olevan tiedon. JAD-tiedosto toimii ohjaustiedostona sovelluksen asennuksessa, ja se sisältää kaiken sovelluksen asennuksessa tarvittavan tiedon. JAD-tiedosto on käytännössä pelkkä tekstitiedosto, joka määrittelee sovelluksen ominaisuuksia kuten esimerkiksi missä näyttömoodissa (vaakanäyttö, pystynäyttö) sovellus suoritetaan. JAD-tiedostoon tehtyjen määrittelien ja ohjauskomentojen avulla matkapuhelin asentaa sovelluksen JAR-tiedostosta (Mahmoud 2002).

## 4.6 WRT-sovellusversio

### 4.6.1 Lähtötilanne

Aloitin WRT-version toteutuksen Java ME -sovellusversion jälkeen tammikuussa 2011. Vaikka WRT-sovelluksien toteutus oli minulle entuudestaan täysin tuntematon, olivat siinä käytettävät ohjelmointikielet (HTML, CSS, JavaScript, Flash Lite) useiden verkkosivuprojektien kautta entuudestaan tuttuja. Ennen sovelluksen toteuttamista opettelini WRT-ohjelmointiympäristön toimintaperiaatteet ja mahdollisuudet Nokian kehittäjäkirjastojen<sup>1</sup> avulla. Nokian ylläpitämissä kehittäjäkirjastoissa ja niiden keskustelupalstoilla on paljon hyviä oppaita ja käytännön vinkkejä ohjelmointiympäristön käyttöön ja hallintaan. Kehittäjäkirjasto ja sitä ylläpitämä Forum Nokia -verkkopalvelu on eräänlainen virtuaalinen kokoontumispaikka ohjelmistoympäristöä käyttäville harrastelijoille ja ammattilaisille.

### 4.6.2 Sovelluksen rakenne

WRT-sovellukset koostuvat erilaisista verkkojulkaisuissa käytetyistä tiedostomuodoista. Vuonislahden kyläkävely -sovelluksen WRT-versiossa käytin yhden HTML-tiedoston kokoonpanomallia, jossa koko sovelluksen rakenne on määritelty yhdessä index.html-tiedostossa (ks. liite 3). HTML-tiedoston lisäksi sovellus koostuu ulkoasua määrittelevistä CSS-tyylitiedostoista, JavaScript-komentotiedostoista ja -rajapinnoista sekä sovelluksen plist-määrittelytiedostosta. WRT-toteutuksissa sisältöjä voidaan jakaa Java ME -version tapaan 3G-verkon välityksellä toistettavaksi, mutta ohjelmointiympäristön hyvän suorituskyvyn vuoksi en nähnyt sitä tarpeelliseksi.

WRT-sovellusversio koostui suunnitelmien mukaisesti etusivusta, kartta- ja infosuivusta sekä erillisistä sisältömateriaalisivuista. Käytän sovelluksen osioista termiä sivu, vaikka sen rakenteessa ei erillisiä sivuja ja niitä kuvaavia HTML-tiedostoja olekaan. Toisin kuin verkkosivujen rakenteissa, sovelluksessa kaikki osiot sijaitsevat yhden HTML-rakennetiedoston sisällä. Sovelluksen osiot on eritelty HTML-tiedostoon div-elementtien id-komennoilla. Käyttämäni jQuery

---

<sup>1</sup> Nokia Web Developer's Library 2.0. Forum.Nokia 2010. <http://library.forum.nokia.com/>.

Mobilen ulkoasu tarvitsee lisäksi jokaiselle erilliselle osiolla data-role-komennon ulkoasun asettelua varten. Kuviossa 9 on havainnollistettu sovelluksen HTML-koodin rakenne, ja siinä näkyvät myös osiot toisistaan erottavat id-nimityskomennot ja ulkoasun käyttämät data-role-komennot.

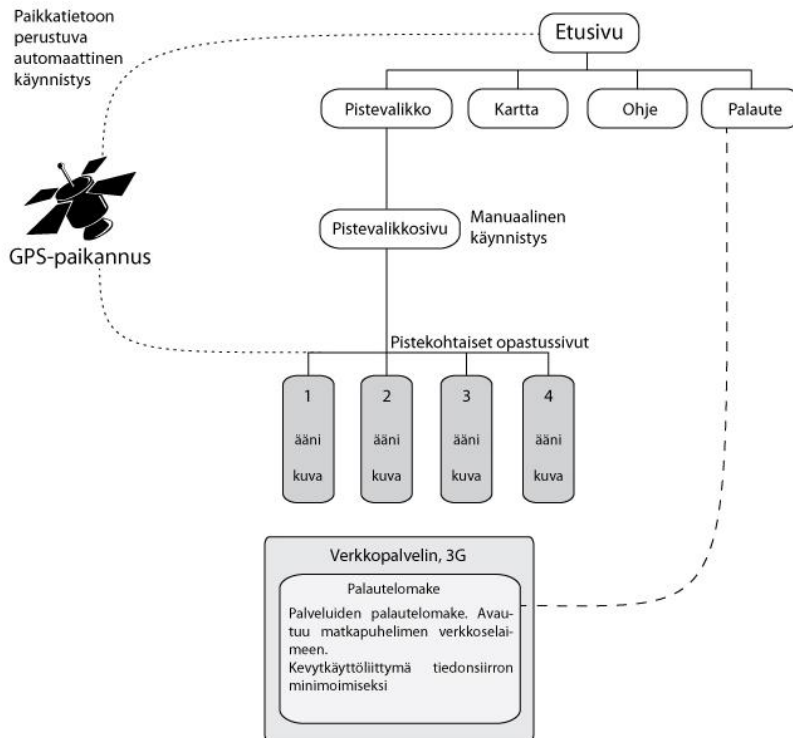
```

<body>
  <div data-role="page" id="etusivu">
    ETUSIVUN ALUE ALKAA
    <div data-role="header" data-position="inline">
      ETUSIVUN YLÄPALKIN SISÄLTÖ
    </div>
    <div data-role="content" id="content" class="ui-body-c">
      ETUSIVUN SISÄLTÖ
    </div>
    <div data-role="footer">
      ETUSIVUN ALAPALKIN SISÄLTÖ
    </div>
    ETUSIVUN ALUE LOPPUU
  </div>

  <div data-role="page" id="piste1">
    OPASTUSPISTEEN 1 ALUE ALKAA
    <div data-role="header" data-position="inline">
      OPASTUSPISTEEN 1 YLÄPALKIN SISÄLTÖ
    </div>
    <div data-role="content" id="content" class="ui-body-c">
      OPASTUSPISTEEN 1 SISÄLTÖ
    </div>
    <div data-role="footer">
      OPASTUSPISTEEN 1 ALAPALKIN SISÄLTÖ
    </div>
    OPASTUSPISTEEN 1 ALUE LOPPUU
  </div>
</body>

```

Kuvio 9. WRT-sovellusversion HTML-koodin rakenne.



Kuvio 10. WRT-sovellusversion rakenne ja toimintamalli.

WRT-sovellusversiossa kaikki sisältö oli sovelluksen sisältä ladattavaa, joten sovelluksen rakenne oli hieman erilainen kuin Java ME -versiossa (ks. kuvio 10). Kuten kuvio 10 ilmenee, olivat suurimmat rakenteelliset muutokset sovelluksen valikkorakenteissa etusivulta pistekohtaisille opastussivuille siirryttäessä. Sovelluksen valikkorakenteet olisi ollut mahdollista toteuttaa samalla tavalla kuin Java ME -versiossa, mutta ulkoasun muutoksien ja paremman suorituskyvyn vuoksi päädyin kuvion mukaiseen ratkaisuun.

#### 4.6.3 Ulkoasu

Toteutin WRT-sovellusversion täysin käytettävään päätelaitteeseen mukautuvan (skaalautuvan) ulkoasun. Skaalautuvuuden vuoksi sovelluksen navigointipainikkeille oli varattava lisää pystysuuntaista tilaa niiden leveyden ollessa päätelaitteen näytön suhteen mukautuvia. Koska navigointipainikkeet tarvitsivat enemmän pystysuuntaista tilaa, päätin siirtää opastuspisteiden manuaalisen käynnistysvalikon erillisen pistevalikkosivun alaisuuteen. Ratkaisua puolsi myös hyvä suorituskyky sovelluksen eri osioiden välisissä siirtymissä, joten erillisen pistevalikon käyttö ei aiheuttanut ylimääräistä viivettä Java ME -versioon verrattuna. Navigointipainikkeiden tilantarpeen takia siirsin myös opastusalueen kar-

tan oman osionsa alle, koska kartan koko olisi etusivulle jätettynä jäänyt liian pieneksi. Havainnekuvat sovelluksen WRT-versiosta ovat opinnäytetyön liitteenä (liite 4).

Käytin sovelluksessa jQuery Mobilen tarjoamaa erityisesti kosketuspuhelimille optimoitua rajapintaa sovelluksen navigointivalikoiden ja yleisilmeen pohjana. JQuery Mobilen tarjoaman rajapinnan avulla sovelluksen ulkoasu ja siirtymät eri osioiden välillä pystytään toteuttamaan valmiiden mallien pohjalta. JQuery Mobilen ulkoasu on valmiiksi tehty päätelaitteiden mukaan skaalautuvaksi, joten valmista pohjaa käyttämällä välttyin erilliseltä skaalautuvuuden määrittelyltä. Rajapinta ja valmiit ulkoasumallit ovat toteuttajan muokattavissa sovelluksen tarpeiden mukaan (The jQuery Project 2010).

#### **4.6.4 Platform Services API**

WRT-tekniikalla toteutettu sovellus yhdistetään puhelimen sisäisiin ominaisuuksiin erillisillä JavaScript-rajapinnoilla. WRT-sovelluksissa rajapinta (API) on JavaScript-kieltä sisältävä tekstitiedosto, joka toimii koneistona puhelimen ja sovelluksen välisessä tietojen välityksessä. Rajapinta sisältää komentorakenteita puhelimen ominaisuuksien käyttöönottoon ja informaation hakemiseen. Nokian tarjoama rajapinta Platform Services API mahdollistaa esimerkiksi paikkatiedon ja puhelimen sensoreiden käyttöönoton HTML-pohjaisissa sovelluksissa (Nokia 2010).

Vuonilahden kyläkävely -sovelluksen WRT-versiossa puhelimen GPS-paikkatieto haetaan Nokian Platform Services 2.0 -rajapintaa käyttämällä. Rajapinta yhdistetään sovellukseen HTML-tiedoston välityksellä. Rajapinnassa on erilaisia komentoja, joiden avulla puhelimen ominaisuudet kutsutaan sovellukseen käytettäväksi. Esimerkiksi GPS-sirun tarjoama paikkatieto kutsutaan Platform Services 2.0 -rajapintaan "nokia.device.load("geolocation");" -komennolla. Täydellinen listaus Platform Services 2.0 -rajapinnan kutsukomennoista löytyy Nokian kehittäjäkirjastosta.

Toteutin GPS-informaation haun sovelluksen etusivun yhteyteen samaan tapaan kuin Java ME -sovellusversiossa. Sovelluksen HTML-tiedostossa on määritelty opastuspisteiden raja-arvot, ja kun sovelluksen seuraamat sijaintikoor-

dinaatit menevät raja-arvojen sisään, käynnistyy koordinaatteihin yhdistetty pistesältösivu (ks. kuvio 11). Käytin sovelluksen GPS-tiedon hakutoiminnossa Nokian kehittäjäkirjaston mallitiedostoissa käytettyä rakennetta, jossa komennot jaetaan useiden olioiden (function) alaisuuteen. GPS-tiedon hakutoiminnon tärkein elementti on kuvion 11 lopussa esitetty puhelimen GPS-toiminnon aktivoiva geolocation-komento. Ilman Platform Services 2.0 -rajapinnassa määriteltyä geolocation-komentoa sovellus ei saa yhteyttä puhelimen GPS-toimintoihin. Asetin GPS-tiedon haulle lisäksi enimmäismääräajan. Mikäli sovellus ei saa minkäänlaista GPS-informaatiota puhelimesta 30 sekunnin kuluessa käynnistymisestä, sovellus ilmoittaa käyttäjälle virhesanoman ja jatkaa toimintaa ilman GPS-informaatiota. Asetettu 30 sekunnin määräaika oli testauksessa käytettävillä Nokian N97 ja N8 -matkapuhelimille riittävä, mutta laajemmin eri päätelaitteille julkaistaessa voisi määräajan nostaminen olla tarpeellista. Määräajan avulla sovellus ei turhaan etsi GPS-informaatiota esimerkiksi puhelimista, joissa GPS-toimintoa ei ole tai joissa GPS:n toiminta on jostain syystä estynyt.

```

<script language="javascript">

    var so = null;
    var tid = ""
    //Toiminto kun sijaintitiedot päivittyvät.
    function onLocationUpdate(position){
        var lon = position.coords.longitude;
        var lat = position.coords.latitude;
            document.getElementById("latit").innerHTML = position.coords.longitude;
            document.getElementById("longi").innerHTML = position.coords.latitude;
    //Opastuspiste 1 koordinaattiseuranta ja käynnistys.
        if (position.coords.longitude > 29.98687 && position.coords.longitude < 29.98891 &&
            position.coords.latitude > 63.147728 && position.coords.latitude < 63.128809){

            window.location = 'index.html#piste1'
        }
    //Opastuspiste 2 koordinaattiseuranta ja käynnistys.
        else if (position.coords.longitude > 29.97971 && position.coords.longitude < 29.98208 &&
            position.coords.latitude > 63.152360 && position.coords.latitude < 63.153354){

            window.location='index.html#piste2'
        }
    };
    //Puhelimen GPS-toiminnon aktivointi ja enimmäisaika GPS-yhteyden hakemiseen mikäli yhteyttä ei saada.
    try{
        so = com.nokia.device.load("", "com.nokia.device.geolocation");
        var tid = so.watchPosition( onLocationUpdate, onLocationError,{timeout:30000});
    }
    </script>

```

Kuvio 11. Sovelluksen yhdistäminen puhelimen GPS-informaatioon Nokia Platform Services 2.0 -rajapinnan avulla.

#### 4.6.5 Sisältömateriaalien toistaminen

WRT-sovelluksissa sisältömateriaaleja pystytään toistamaan joko suoraan HTML-koodiin upotettuina elementteinä tai erillisten JavaScript-toistimien välityksellä. JavaScript-toistimen avulla sisältömateriaaleihin saadaan tuotua dynaamisuutta, ja se soveltuukin paremmin useamman sisältöelementin toistimeksi. HTML-elementteinä upotettu sisältö on suorituskykyisempää, koska sisältö toistuu suoraan ilman erillisiä toistimia. Käytin Vuonislahden kyläkävely-sovelluksen kuvasisällöissä erillistä JavaScript-toistinta, koska useammasta kuvatiedostosta koostuvan kuvaesityksen toteutus oli sitä kautta yksinkertaisempaa ja visuaalisesti miellyttävämpää. Kuvasisältöjen toistaminen olisi ollut mahdollista myös HTML-elementteinä, mutta valmista JavaScript-toistinta käyttämällä vältin ylimääräisen ohjelmointityön. Käytin kuvien esittämisessä Translucent Slideshow<sup>1</sup> -nimistä valmista kuvatoistinta. Toistin on toteutettu JavaScript-kielellä ja soveltuu hyvin useista kuvatiedostoista koostuvien kuvaesitysten toistamiseen. Kuvatoistimen ominaisuudet kuten kuvien esityskoot ja vaihtovälit ovat täysin muokattavissa toteutuksen tarpeiden mukaisesti. Liitin kuvatoistimen mukana tulleen JavaScript-komentotiedoston sovelluksen HTML-rakennetiedostoon ja määrittelin toistettavat sisällöt sovelluksen tiedostokansioista. Muokkasin kuvatoistimesta sen koon vastaamaan puhelimen pystysuuntaisen näytön leveyttä.

Äänitiedostojen toistamiseen käytin Nokian kehittäjäkirjastoista löytämäni malliratkaisua, jossa käytetään HTML-elementin ja JavaScript-toistimen yhdistelmää<sup>2</sup>. Toistimessa käytetään HTML-koodiin upotettuja JavaScript-komentoja, joiden avulla määritetään opastuspisteen sivulle upotettava embed-komento. Nokian kehittäjäkirjastoissa olevassa mallitoistin on toteutettu yhtä äänitiedostoa varten, joten jouduin muokkaamaan toistimen tukemaan useamman äänitiedoston toistamista. Toteutin toistimen siten, että opastuspiste-osion auetessa JavaScript-komennot tekevät HTML-koodiin äänitoistimen ja siihen liitetyn äänitiedoston embed-komentoa käyttäen. Tein jokaiseen opastuspiste-osioon lisäksi äänen käynnistys- ja pysäytyspainikkeet. Pysäytyspainike poistaa JavaScript-

---

<sup>1</sup> Translucent Slideshow Script. Dynamic Drive 2011. <http://www.dynamicdrive.com/dynamicindex14/translucentslide.htm>.

<sup>2</sup> Audio in S60 WRT. Nokia 2011. [http://wiki.forum.nokia.com/index.php/Audio\\_in\\_S60\\_WRT](http://wiki.forum.nokia.com/index.php/Audio_in_S60_WRT).

komentojen tekemän äänitoistimen embed-komennot ja käynnistuspainike vastaavasti tekee embed-komennon uudestaan.

#### **4.6.6 Julkaisu ja jakelu**

Nokian WRT-sovellusten julkaisussa ja jakelussa käytetään WGZ-tiedostomuotoa. WGZ on pakattu tiedostomuoto, joka sisältää kaiken sovelluksessa olevan paikallisen datan. Useista eri tiedostomuodoista koostuva WRT-sovellus voidaan pakata WGZ-tiedostomuotoon ZIP-tiedostoja tekevillä pakkausohjelmilla sekä esimerkiksi Adobe Dreamweaveriin saatavalla WRT-laajennuksella. ZIP-pakkauksen kautta tehtävä WGZ-tiedosto pakataan normaalisti ZIP-tiedostomuotoon, minkä jälkeen tiedostopäätte vain vaihdetaan ZIP:stä WGZ:ksi. WGZ-pakkaus pienentää sovelluksen tiedostokokoa pakkaamalla sovelluksen tarvitsemat tiedostot yhdeksi WGZ-tiedostopakettiksi. WGZ-sovelluksien asennustoiminto käynnistyy automaattisesti, kun tiedosto avataan sitä tukevissa puhelinmalleissa.

WGZ-pakattu sovellus tarvitsee toimiakseen vähintään sovelluksen minimikoonpanon eli yhden HTML-tiedoston ja info.plist-määrittystiedoston. Info.plist-tiedostossa määritellään sovelluksen perustiedot, kuten nimi, tekijä ja versio-numero (ks. kuvio 12). Lisäksi plist-tiedostossa määritellään sovelluksen käynnistyksen yhteydessä avattava tiedosto, joka esimerkiksi Vuonislahden kyläkävely -sovelluksessa oli rakennetiedosto index.html. Lisäksi plist-tiedostossa määritellään voidaanko sovelluksesta käynnistää pienoishjelma (mini-widget) puhelimen työpöydällä näytettäväksi. Plist-määrittystiedoston tulee olla Nokian määrittelemän kieliopin mukainen. Malli plist-tiedoston rakenteesta löytyy Nokian kehittäjäkirjastoista. (Nokia 2011)



```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Nokia//DTD PLIST 1.0//EN" "http://www.nokia.com/DTDs/plist-1.0.dtd">
<plist version="1.0">
  <dict>
    <key>DisplayName</key>
    <string>Vuonislahden Kyläkävely</string>
    <key>Identifier</key>
    <string>com.Kyläkävely.basic.widget</string>
    <key>Version</key>
    <string>4.2</string>
    <key>MainHTML</key>
    <string>index.html</string>
    <key>AllowNetworkAccess</key>
    <true/>
    <key>MiniViewEnabled</key>
    <false/>
  </dict>
</plist>
```

Kuvio 12. Vuonislahden kyläkävely -sovelluksen WRT-version info.plist-määrittystiedosto.

## 5 TULOKSET

### 5.1 Soveltuvuus

Toteutin Vuonislahden kyläkävely -palvelunimikkeellä kaksi matkapuhelinsovellusta. Sovellukset ovat keskenään hyvin erilaisia pohjaratkaisuiltaan ja toimintaperiaatteiltaan. Java ME -version toiminta perustuu taustalla suoritettavaan virtuaalikoneeseen, kun taas WRT-versio perustuu yhdistelmään suoraan puhelimessa suoritettavista koodikielistä. Molemmat sovellusversiot toimivat moitteettomasti laajennetun todellisuuden palveluina, vaikkakin niiden käytettävyydessä oli huomattavia eroja. Koska palvelun sisältöinä oli huomattavan raskaita ääni- ja kuvatiedostoja, olivat suorituskyky ja sitä kautta käytettävyys tärkeimpiä vertailtavia ominaisuuksia.

Kun toteutettuja sovellusversioita verrataan aiempiin Pohjois-Karjalassa navigaattoreille toteutettuihin opastuspalveluihin, on matkapuhelin päätelaitteena huomattavasti miellyttävämpi ja selkeämpi käyttää. Koska sovelluksen rakenne on täysin toteutuksen mukaan suunniteltavissa, pystytään myös käytettävyys säilyttämään yksinkertaisena. Kummassakaan matkapuhelinsovelluksessa ei ilmennyt aiemmissa navigaattoritoteutuksissa havaitsemiani käytettävyys- tai päätelaitteongelmia. Toteutettujen sovellusten vertailu aiempiin navigaattoritoteutuksiin on hankalaa, sillä teknologia ja päätelaitteet ovat kehittyneet toteutusten julkaisuajankohtien välillä paljon. Olen kuitenkin tulosten myötä hyvin vakuuttunut siitä, että matkapuhelin soveltuu navigaattoria paremmin laajennetun todellisuuden palveluiden tarjoamisalustaksi.

Keskityn seuraavissa kappaleissa käsittelemään sovellusversioiden keskinäistä vertailua ja saavutettuja tuloksia, sillä versioiden välinen vertailu on mielestäni huomattavasti navigaattoritoteutuksiin vertaamista relevantimpaa. Keskityn sovelluksien vertailussa laajennetun todellisuuden palvelua ajatellen tärkeisiin teknisiin ominaisuuksiin. Sovelluksien vertailussa käytössäni oli kaksi erilaista Nokian kosketusnäyttöpuhelimta: Symbian<sup>1</sup> (S60 5th.) -käyttöjärjestelmää käyttävä N97 ja Symbian<sup>3</sup> -käyttöjärjestelmää käyttävä N8. Molemmat puhelimet käyttävät sovelluksen suunnittelussakin valitsemaani 640 x 360 pikselin näyttö-

resoluutiota. Valitsin puhelimet niiden käyttämän näyttöresoluution ja molempia sovellusversioita tukevan käyttöjärjestelmän takia.

## 5.2 Prosessi

Prosesseina molempien sovellusversioiden toteutukset olivat keskenään hyvin erilaisia. WRT-toteutus oli prosessina enemmän työtä vaativa, sillä aloittaessani toteutuksen en tuntenut ohjelmointiympäristöä ollenkaan. JM-Mobile Editori tarjosi Java ME -sovellusversioon enemmän valmiita ratkaisuja käytettäväksi, kun taas WRT-toteutuksessa jouduin etsimään ratkaisut pääasiassa itse. Vaikka WRT-sovellusversio oli työmäärältään vaativampi, antoi se kuitenkin enemmän vaihtoehtoja sovelluksen toteutukseen, koska siinä ei ollut JM-Mobile Editorin kaltaista toteutusohjelmaa käytettäviä ominaisuuksia rajoittamassa.

## 5.3 Työkalut

Sovelluksien toteutuksissa käytetyt ohjelmat ovat keskenään hyvin erilaisia ja osittain myös eri käyttötarkoituksiin suunniteltuja, joten niiden keskinäinen vertailu on hankalaa. Siinä missä Adobe Dreamweaver on tarkoitettu monipuoliseen verkkosivu- ja sovellustoteutukseen visuaalista käyttöliittymää ja tekstieditoria yhdistelemällä, on JM-Mobile Editor rajoittunut visuaalinen editori pelkästään Java ME -ohjelmointiympäristön sovelluksien toteutukseen. Sovelluksen toteutus on huomattavasti helpompaa JM-Mobile Editorilla, sillä se ei vaadi GPS-tiedon yhdistämistä lukuun ottamatta ohjelmointiosaamista. Kuitenkin Dreamweaverilla toteutetuissa WRT-sovelluksissa on paljon enemmän valinnanvaraa niin ominaisuuksien kuin rakenteidenkin osalta. Nokian kehittäjäkirjastojen tarjoamat ohjeet ja oppaat helpottavat WRT-sovelluksiin ja Dreamweaverin käyttöön tutustumista. Lisäksi Adobe Dreamweaver on JM-Mobile Editoriin verrattuna huomattavasti suuremman ohjelmistoyrityksen tarjoama tuote, joten päivityvyys ja tuotetuki ovat paljon parempia.

## 5.4 Rakenne

Suunnitteluvaiheessa tekemieni käytettävyys- ja rakennelinjauksien toteutus onnistui molemmissa sovellusversioissa hyvin. Sain toteutettua sovelluksien rakenteet pieniä muutoksia lukuun ottamatta suunnitelmien mukaisesti. Java ME -sovellusversiossa rakenteellisia muutoksia aiheutti heikko suorituskyky, joten muutosten tekeminen oli käytettävyyden kannalta pakollista. WRT-versiossa tehdyt muutokset taas johtuivat käyttämästäni ulkoasupohjasta, koska opastuspiste-valikon siirtäminen omaan osioonsa paransi sovelluksen käytettävyyttä. WRT-versiossa tekemäni ratkaisu ei olisi ollut välttämätön, mutta sen avulla sovelluksen etusivu pysyi käyttäjälle selkeämpänä ja siten helpommin käytettävänä.

Sovelluksien rakenteiden osalta mielestäni WRT-versio oli paremmin laajennettun todellisuuden palveluihin soveltuva. WRT-toteutuksessa rakenteen toteutus on paljon vapaampaa kuin Java ME -versiossa. Koska WRT-sovelluksissa toiminnot jaetaan erilaisia ohjelmointikieliä käyttävien tiedostomuotojen alle, on sovelluksen eri toimintojen samanaikainen suorittaminen huomattavasti kevyempää. Esimerkiksi JavaScript-rajapinnoilta käytettävä puhelimen ominaisuuksien lataaminen ei vaikuta hidastavasti CSS-tiedostosta tapahtuvan ulkoasun lataamiseen. JM-Mobile Editorilla toteutettujen sovellusten heikko suorituskyky rajoittaa sovellusrakenteiden suunnittelua ja esimerkiksi Vuonislahden kyläkävely -toteutuksessa jouduin muuttamaan sovellusrakennetta ohjelmointiympäristön suorituskyvyn takia. Erillisistä koodikielistä koostuvan WRT-sovelluksen päivitettävyys on myös huomattavasti Java ME-sovellusta helpompaa, sillä esimerkiksi ulkoasun päivittämiseen riittää pelkkä CSS-tiedoston muokkaaminen.

Jatkokehitystä ajatellen WRT-ohjelmointiympäristön etuna on myös sen yleisesti verkkototeutuksissa käytössä olevat koodikielet. Niiden avulla esimerkiksi internetsisältöjen yhdistäminen sovellukseen tai sovelluksen muuntaminen täysin verkossa toimivaksi on helpompaa. Vaikka myös Java ME -sovellukseen on mahdollista tuoda erillisiä verkkosisältöjä, on jo valmiiksi verkkosisällöissä käytettäviä koodikieliä sisältävä WRT parempi vaihtoehto.

## 5.5 Suorituskyky

Sovelluksien suorituskyvyissä oli merkittäviä eroja, jotka osaltaan vaikuttivat myös sovelluksen käytettävyyteen. Toteutettu Java ME -versio oli toiminnaltaan huomattavasti WRT-versiota hitaampi. Java ME -version hitaus johtui sen käyttämästä virtuaalikoneesta ja raskaiden sisältöjen aiheuttamasta kuormituksesta.

Vuonislahden kyläkävely on laajennetun todellisuuden matkapuhelinpalvelu ja yksi sen tärkeimmistä ominaisuuksista on käytettävyys, johon huono suorituskyky vaikuttaa kielteisesti. Sovellusversioista WRT oli huomattavasti miellyttävämpi käyttää, sillä sen toiminta oli sovelluksen kaikissa osioissa ja opastuspisteiden sisältöjen koosta riippumatta paljon Java ME -sovellusversiota suorituskykyisempää. Esimerkiksi painikkeiden reagoitaviiveet olivat Java ME -versiossa helposti huomattavissa, kun taas WRT-versiossa painikkeet reagoivat käyttäjän toimintaan välittömästi. Vaikka kyseessä on pieniä käytettävyyteen vaikuttavia asioita, on niillä suuri merkitys ohjelmointiympäristön soveltuvuudessa laajennetun todellisuuden palveluun käytettäväksi.

Vertailin sovelluksien suorituskykyä kahden osion välisen siirtymäajan perusteella (ks. taulukko 1). Vertailu toteutettiin Nokia N97 -puhelinmallia käyttämällä. Aikamittaus alkoi, kun sovelluksen navigointipainiketta painettiin, ja päättyi, kun ensimmäisen opastuspisteen sivu oli kokonaan latautunut. Sisältömateriaalina ensimmäisen opastuspisteen sivulla oli yksi äänitiedosto (758 kilotavua) ja neljä kuvatiedostoa (yhteensä 187 kilotavua).

Testissä WRT-sovellusversio osoittautui huomattavasti suorituskykyisemmäksi, vaikka siinä etusivulta opastuspisteiden sivuille siirrytään erillisen pistevalikon kautta. Käytettävyyden kannalta hyvänä pidettävän latausviive verkkosivuilla on enintään kahdeksan sekuntia. Kymmenessä sekunnissa käyttäjä turhautuu koikelemaan samaa toimintoa uudestaan ja usein poistuu palvelusta. (Jakob Nielsen 2010.) Vaikka verkkosivut ja matkapuhelinsovellukset ovat toimintaperiaatteiltaan erilaisia, ovat niiden käyttäjien asettamat odotukset käytettävyyden suhteen mielestäni vertailukelpoisia.

Taulukko 1. Sovellusversioiden latausajat osioiden välisissä siirtymissä.

Sovellusversio	Siirtymäaika etusivu – 1. opastuspiste*	Siirtymäaika 1. opastuspiste – etusivu*
Java ME (paikallinen)	13 s.	11 s.
Java ME (verkkora- kenne)	7 s.	11 s.
WRT	4 s.	2 s.

\*Vertailu suoritettiin Nokia N97 puhelinmallilla 11.3.2011. Vertailuajat on mitattu digitaalista sekuntikelloa käyttämällä, joten mitatut ajat ovat ainoastaan viitteellisiä. Mittauksen tarkkuus  $\pm 1$  s.

## 5.6 Sisältömateriaalin toistaminen

Sisältömateriaalien toistamisessa ohjelmointiympäristöt erosivat toisistaan merkittävästi. Java ME -sovellusversiossa sisältöjen toistamiseen on tarjolla ainoastaan JM-Mobile Editorin toistimet, kun taas WRT-versiossa käytettävien toistinten valikoima on huomattavasti laajempi. JM-Mobile Editorin tarjoamat toistimet eivät soveltuneet raskaiden sisältöjen toistamiseen, sillä sisältömateriaalien toistaminen heikensi koko sovelluksen suorituskykyä. Vaikka kuvasisältöjen siirtäminen verkkopalvelimelta toistettaviksi paransi suorituskykyä, oli se silti WRT-versioon verrattuna hidasta. WRT-versiossa käyttämäni toistimet olivat suorituskyvyltään hyviä ja sisällöt toistuivat paikallisesti puhelimen muistista ongelmitta. WRT:n mahdollistamat useat vaihtoehtoiset JavaScript- ja HTML-toistimet ovat sovelluskehitystä ja -kehittäjiä ajatellen huomattavasti JM-Mobile Editorin toistimia parempi ratkaisu.

## 5.7 Ulkoasu

Toteutettujen sovellusversioiden ulkoasut olivat keskenään hyvin erilaisia. Koska kyseessä oli teknisinä pilotteina toimivat sovellusversiot, jätiin niiden ulkoasujen suunnittelun vähemmälle huomiolle. Ulkoasujen osalta keskityn vertailemaan sovellusversioiden skaalautuvuutta eli ulkoasun mukautumista eri kokoisten näyttöjen mukaisesti (ks. kuvio 13). Käytin testauksessa Nokia N97 -puhelinmallia ja testasin sovelluksen toimivuutta vaak- ja pystynäyttötilassa.

Otin testausta varten Java ME -sovellusversioon tekemäni vaakasuuntaisen näytön lukituksen pois käytöstä, jotta sovelluksen havainnointi onnistuisi myös pystysuuntaisessa näytössä.



Kuvio 13. Sovellusversioiden ulkoasun skaalautuvuus eri näyttötiloissa. Tarkeimmat havainnekuvat sovelluksen ulkoasusta ovat liitteissä 4 ja 5.

JM Mobile Editorilla toteutetut sovellukset on mahdollista asettaa skaalautumaan käytettävän laitteen näytön mukaiseksi. Kuten kuvioista 13 ilmenee, ei skaalautuvuus toimi käytännössä ollenkaan, vaan sovelluksesta on pystysuuntaisessa näytössä näkyvissä ainoastaan sen leveyteen (360 px) mahtuva osa. WRT-versiossa sisällön skaalaus oli täysin sujuvaa ja vaakasuuntaisessa näyttötilassa sisältö muuttui vertikaalisesti vieritettäväksi.

## 5.8 Ominaisuudet

Paikkatiedon yhdistäminen tarjottavaan sisältöön onnistui molemmissa sovellusversioissa hyvin. Puhelimen sisäisten ominaisuuksien käyttö on tuettu molemmissa teknologioissa laajennetun todellisuuden palveluita ajatellen riittävästi. Kuitenkin Java ME -versiossa käytettävissä olevat ominaisuudet ovat huomattavasti rajoittuneemmat. Siinä missä WRT:n Nokia Platform Services -rajapinnalla voidaan tietoa hakea lähes kaikista saatavilla olevista sensoreista ja toiminnoista, on JM-Mobile Editorilla käytössä ainoastaan puhelimen aika- ja

sijaintitiedot. GPS-paikannuksen käyttö oli molemmissa sovellusversioissa hyvin suorituskykyistä ja sijaintimäärityksen tarkkuus puhelimen GPS-sirun suorituskyvyn mukaista. Vuonislahden kyläkävely -sovelluksessa pelkkä paikkatieto oli vielä riittävä ominaisuus, mutta WRT:n tarjoama laajempi puhelimen muiden sensoreiden ja ominaisuuksien tuki tekee siitä jatkokehitystä ajatellen huomattavasti Java ME:tä paremman vaihtoehdon.

## 5.9 Yhteensopivuus

Testaukseen valitsemani päätelaitteet olivat molempien ohjelmointiympäristöjen kanssa yhteensopivia. Vaikka toteutetun palvelun tavoitteena ei ollut laaja yhteensopivien matkapuhelinten määrä, on yhteensopivuuden vertaaminen palveluiden jatkokehitystä ja julkaisua ajatellen tärkeää. Mitä laajemmin palvelussa käytetty ohjelmointiympäristö on yhteensopiva erilaisten puhelinmallien kautta, sitä laajempi käyttäjäkunta sillä voidaan saavuttaa.

Yhteensopivien matkapuhelinmallien määrässä Java ME on huomattavasti WRT:tä monipuolisempi vaihtoehto. Java ME -ohjelmointiympäristö on yhteensopiva yli 600 eri puhelinmallin kanssa (Java User Group 2010). Vaikka yhteensopivien puhelinmallien määrä on suuri, täytyy kuitenkin huomioida, että laajennetun todellisuuden palveluissa käytettävien sisältöjen esittämiseen niistä soveltuu huomattavasti pienempi osa. Puhelinmallin yhteensopivuuden ja sen suorituskyvyn laajennetun todellisuuden palvelussa määrittelevät puhelimen käyttömuistin (RAM) määrä ja käytettävien sisältöjen tiedostokoot. Java ME -sovelluksissa oleva virtuaalikone lataa sisällöt ensin puhelimen käyttömuistiin ja suorittaa ne sen jälkeen sovelluksessa. Mikäli palvelussa toistettavien sisältöjen tiedostokoot ylittävät käyttömuistin määrän, lopettaa virtuaalikone toimintansa ja sovellus sulkeutuu. Java ME -versiossa käyttämälläni XML-verkkorakenteella sisältöjen toistuvuutta pystytään nopeuttamaan ulkopuolelta ladattavilla sisältömateriaaleilla. Ulkopuolelta ladattavat tiedostot vähentävät sovelluksen suorittamiseen tarvittavaa käyttömuistin määrää, koska ne eivät ole koko aikaa mukana sovelluksen toiminnassa. Mikäli vapaana oleva käyttömuisti ei riitä ulkopuolisten sisältöjen lataamiseen ja toistamiseen, on lopputulos sama ja sovellus sulkeutuu.



WRT-ohjelmointiympäristön kanssa yhteensopivien puhelinmallien määrä on vielä toistaiseksi melko vähäinen. Forum Nokian mukaan WRT-sovellukset toimivat Nokian S40 ja S60 3rd Edition ja 5th Edition (Symbian^1) -käyttöjärjestelmissä sekä uudemmassa Symbian^3-käyttöjärjestelmässä. Kyseisillä käyttöjärjestelmillä varustettuja yhteensopivia puhelinmalleja on Nokian mukaan 58 kpl. (Nokia 2011.)

WRT-ohjelmointiympäristössä toteutetut sovellukset on mahdollista muuntaa yhteensopiviksi myös muita käyttöjärjestelmiä käyttävien puhelinten kanssa. Yksi muuntamistapa on esimerkiksi PhoneGap-rajapinta, joka toimii käyttämäni Platform Services -rajapintaa vastaavalla tekniikalla ja on yhteensopiva useampien käyttöjärjestelmien kanssa. PhoneGap-rajapinnalla WRT-kokoonpanolla toimivat sovellukset on mahdollista saada yhteensopiviksi Apple iOS-, Android-, Palm- ja Windows Mobile -käyttöjärjestelmien kanssa. PhoneGapin kaltaista universaalia rajapintaa käyttämällä yhteensopivien puhelinmallien määrää saadaan kasvatettua huomattavasti. (Nitobi 2011.)

Vaikka Java ME -ohjelmointiympäristön kanssa yhteensopivien matkapuhelinmallien määrä on huomattavasti WRT-toteutuksia laajempi, on suorituskykyisten ja raskaiden sisältöjen toistamiseen soveltuvien puhelinmallien määrä molemmilla lähes sama. WRT:n laajentaminen eri käyttöjärjestelmiin universaaleja rajapintoja käyttämällä tuo yhteensopivien puhelinmallien määrän jo hyvin lähelle Java ME:n määrää.

## 6 POHDINTA

### 6.1 Tulosten yhteenveto

Tavoitteena opinnäytetyössäni oli toteuttaa pilottisovellukset kahta eri ohjelmointialustaa käyttäen Vuonislahden kyläkävely -mobiilipalveluun. Tavoitteena oli myös vertailla käytettävien ohjelmointialustojen soveltuvuutta laajennetun todellisuuden elämyspalveluiden toteuttamiseen. Valitsin vertailun pohjaksi aiemmin tutkimani laajennetun todellisuuden palveluiden määritteet, joiden mukaan laajennetun todellisuuden palveluissa sisällöllä ja sen esittämistavalla on suuri vaikutus realismin rikastamisessa. Määritteiden mukaan palvelu tulee olla rakennettu siten, että sisällön esiintuovat mekanismit pidetään mahdollisimman näkymättömissä. Aiemmissa navigaattoritoteutuksissa laitteiden huono käytettävyys vaikutti olennaisesti palveluiden osittaiseen epäonnistumiseen. Toteutettujen sovellusversioiden vertailussa keskityin pääasiassa sisältöjen esittämiseen ja suorituskyykyyn, koska mielestäni ne ovat tärkeimmät ominaisuudet laajennetun todellisuuden palveluun sopivuuden mittaamiseen. Koska toimintani painottui pääasiassa sovellusten tekniseen toteutukseen, keskityin ohjelmointiympäristöjen vertailussa ainoastaan niiden tekniseen soveltuvuuteen.

Opinnäytetyössä tehdyn vertailun tuloksena käytetyistä ohjelmointirajapinnoista WRT on selkeästi parempi laajennetun todellisuuden palveluissa käytettäväksi. WRT:n monipuolisuus ja sen Java ME:tä huomattavasti parempi suorituskyyky tekee siitä Java ME:tä paremman alustaratkaisun. Koska laajennetun todellisuuden palveluissa raskaat sisällöt, kuten äänitiedostot ja kuvat, ovat keskeisessä osassa palvelua, on WRT:n suorituskyyky kahden ohjelmointiympäristön vertailussa sen suurin etu. Sisältöjen esittämisessä WRT-sovelluksiin yhteensopivat JavaScript-toistimet mahdollistavat paljon monipuolisemman esitystapojen valikoiman. Laajan toistinvalikoiman vuoksi käyttämäni toistimet eivät välttämättä ole tarjolla olevista kaikkein suorituskyykyisimmät, mutta nekin suoriutuivat vertailussa huomattavasti JM-Mobile Editorin toistimia paremmin.

Jatkokehitystä ajatellen myös WRT:n käyttämät verkkojulkaisuista tutut ohjelmointikielet ovat selkeä etu esimerkiksi sovellusta päivitettäessä. WRT:n mahdollistama sovelluksen ja verkkosisältöjen yhdistäminen tekee päivitystyöstä

vaivatonta niin ylläpitäjille kuin myös sovelluksen käyttäjille. Ylläpitäjät voivat verkkosisältöjen avulla tuoda sovellukseen jatkuvasti päivittyvää sisältöä ilman että käyttäjän tarvitsee ladata tai asentaa sovellusta uudestaan. Sama toimintatapa on toki mahdollista myös Java ME-sovelluksissa, mutta silloin sovelluksessa pitää olla tehtynä erillinen XML-verkkorakenne internetissä sijaitsevia sisältöjä hakemassa.

## 6.2 Vertailun vaihtoehtoinen tulos

Vertailun tulos on hyvin riippuvainen ohjelmointiympäristöjen käyttötarkoituksesta. Toteutettu vertailu oli yksiselitteinen sille asetettujen määritteiden puitteissa, koska vertailussa oli tarkoitus tutkia ohjelmointiympäristöjen soveltuvuutta laajennetun todellisuuden palveluun ainoastaan valituille puhelinmalleille. Vertailun lopputuloksesta olisi mahdollista saada myös hyvin erilainen asetettuja tavoitteita muuttamalla.

Mikäli tavoitteena olisi ollut saavuttaa mahdollisimman laaja toimivuus mahdollisimman monella puhelinmallilla käytettävyydestä ja suorituskyvystä välittämättä, olisi Java ME -ohjelmointiympäristö vertailussa paljon vahvemmassa asemassa. Java ME:n toistaiseksi suurimpana etuna on sen hyvä yhteensopivuus vanhempien puhelinmallien kanssa. Siinä missä käyttämäni WRT-kokoonpanon toiminta rajoittuu toistaiseksi ainoastaan Nokian Symbian S40-, Symbian^1- ja Symbian^3-käyttäjärjestelmille, on Java ME:tä tukevien puhelinmallien ja laitevalmistajien lista moninkertainen.

Toteutettu vertailu ei siis ole mikään universaali totuus, vaan siinä saavutetun lopputuloksen pätevyys on pitkälti riippuvainen palvelulle asetetuista tavoitteista. Toistaiseksi vielä tavoitteita muuttamalla voi vertailun lopputulos kääntyä hyvinkin tasaväkiseksi, mutta on myös huomioitava vertailtujen ohjelmointiympäristöjen täysin toisistaan poikkeava toimintaperiaate. Siinä missä Java ME edustaa vielä vanhempaa ja perinteistä tapaa sovellustoteutuksesta, on WRT taas täysin uudenlainen ja varmasti tulevaisuudessa yhä enemmän yleistyvä ja yhteensopivuudeltaan kehittyvä toteutusmalli.

### 6.3 Käytetyt menetelmät

Opinnäytetyössäni käyttämäni menetelmät sovellusten toteutukseen olivat mielestäni hyvin vertailukelpoisia keskenään. Valitsin ohjelmointiympäristöistä kaksi hyvin erilaista toteutustapaa, joten vertailun lopputulokset olivat myös keskenään hyvin erilaisia. Koska ohjelmointiympäristöjen kirjo on matkapuhelinsovelluksissa hyvin laaja, ei tulosten perusteella pysty sanomaan yksiselitteisesti parasta toteutustapaa. Valituista ohjelmointiympäristöistä WRT:ssä jouduin tekemään valintoja esimerkiksi käytettävässä rajapinnassa ja sisältömateriaalien toistimissa. Vertailun lopputulokseen tekemilläni valinnoilla ei ollut merkitystä, sillä vertailu keskittyi enemmänkin ohjelmointiympäristöjen ominaisuuksiin kuin niissä käytettyihin elementteihin.

Sovellusversioiden keskinäisessä vertailussa keskityin pääasiassa teknisiin ominaisuuksiin. Pidin vertailussa käytettävyyttä ja suorituskykyä sovellusversioiden keskeisinä vertailuelementteinä. Valitsin käytettävyyden ja suorituskyvyn vertailuelementeiksi laajennetun todellisuuden palvelumääritteiden perusteella. Laajennetun todellisuuden palvelun määritteitä tutkiessa selkeästi tärkeimmiksi ominaisuuksiksi nousivat juuri käytettävyys, suorituskyky ja niiden vaikutus realismin rikastamisesta syntyvään elämykseen. Käytettävyyttä ja suorituskykyä vertailemalla pystyin paremmin vertaamaan saavutettuja tuloksia myös aiempiin toteutuksiin ja tarkastelemaan saavutettua kehitystyötä ja ohjelmointiympäristöjen käyttökelpoisuutta valittua palvelumuotoa ajatellen.

### 6.4 Opinnäytetyöni oppimisprosessina

Toteuttamani sovellusversiot olivat minulle ensimmäiset laajennetun todellisuuden matkapuhelinsovellukset. Pystyin käyttämään työskentelyssä hyödyksi aiemmista projekteista saatua tietopohjaa ja kehittämään sitä aiempaa laajemmaksi. WRT-tekniikka oli minulle ennen opinnäytetyöni aloittamista täysin tuntematon, mutta pystyin opinnäytetyöni aikana omaksumaan sen itselleni työkaluksi. Pystyin prosessin aikana syventämään myös Java ME -ohjelmointiympäristön osaamistani.

Opinnäytetyössäni tekemä tutkimus laajennetun todellisuuden palveluihin ja aihetta koskevaan kirjallisuuteen antoi minulle hyvät valmiudet tarkastella ai-

emmin toteutettuja navigaattoripalveluita täysin uudesta näkökulmasta. Prosesin aikana omaksutuilla näkökulmilla pystyin löytämään vanhoista toteutuksista paljon jatkokehittettävää ja jalostamaan tietoja toteutettuihin sovellusversioihin. Opinnäytetyöprosessini toimi siten erinomaisena mahdollisuutena syventää opiskelun aikana saavuttamiani tietoja ja taitoja sekä oppia näkemään toteutettuja palveluita uusista näkökulmista.

Onnistuin opinnäytetyöni tavoitteessa mielestäni hyvin, sillä sain toteutettua kaksi käyttökelpoista sovellusversiota laajennetun todellisuuden palveluissa käytettäväksi. Onnistuin huomioimaan aiemmissa toteutuksissa ilmenneitä ongelmia sekä kehittämään niiden pohjalta palvelumuotoa paremmaksi. Lisäksi löysin käyttämistäni ohjelmointiympäristöistä laajennetun todellisuuden palveluiden kannalta olennaisia vertailukohtia. Tavoitteiden kannalta toteuttamani vertailun lopputuloksen yksiselitteisyys on myös hyvä asia, mutta kuten luvussa 6.2 tekemäni pohdinta osoittaa, voi tulos olla eri vertailukriteereillä katsottuna täysin päinvastainen.

## **6.5 Palveluiden tulevaisuus ja jatkokehitysideat**

Laajennetun todellisuuden mobiilipalvelut kasvavat lähitulevaisuudessa yhä merkittävämmäksi keinoksi sisältöjen välittämiseen. Varsinkin matkailijoiden keskuudessa ne tarjoavat modernin kanavan kertoa tietoa ja tuoda todellisiin ympäristöihin sellaisia sisältöjä, joihin tutustuminen olisi muuten käyttäjille mahdotonta. Palveluiden avulla pystytään rikastuttamaan todellista ympäristöä ja siten tarjoamaan käyttäjille parempi käsitys palvelun kohteesta. Pelkästään Suomi on täynnä kohteita, joissa ennen perintötietona sukupolvilta toiselle kulkeneet tarinat ja tiedot ovat vähitellen katoamassa. Laajennetun todellisuuden mobiilipalveluiden avulla näiden tietojen siirtämistä voidaan jatkaa sukupolvilta seuraaville modernein työkaluin.

Palveluiden päätelaitteina matkapuhelimet ovat tällä hetkellä huomattavasti esimerkiksi navigaattoreita käyttökelpoisempia. Uskonkin, että tulevaisuudessa vastaavanlaiset palvelut toteutetaan pääasiassa matkapuhelimille. Matkapuhelinten laitekannassa tapahtuvan kehityksen myötä WRT:n kaltaiset moderneille matkapuhelimille kehitetyt ohjelmointiympäristöt yleistyvät entistä enemmän.

WRT:n ja universaalien rajapintojen myötä sovelluksia pystytään toteuttamaan entistä monialustaisemmiksi. Ohjelmointiympäristöissä tapahtuva muutos kohti enemmän verkkosivumaisia rakenteita vaikuttaa laajemmin myös palvelurakenteisiin. Esimerkiksi verkkosivuja ja sovelluksia yhdistelevät verkkosovellukset voivat hyvinkin olla seuraava matkapuhelinpalveluiden toteuttamismuoto ja raja erillisten toteutusten väliltä häviää. Verkkosovellusten kautta samoilla toteutuksilla pystytään saavuttamaan niin matkapuhelin- kuin tietokone- ja tablet-laitteiden käyttäjät, joten WRT:n kaltaiset ohjelmointiympäristöt voivat osaltaan vaikuttaa koko verkkopalveluiden suunnitteluun.

Palveluiden tekniset ominaisuudet ovat laitekannan mukaan jatkuvasti kehittyviä, joten keinoja todellisuuden rikastamiseen tulee jatkuvasti lisää. Ääni- ja kuvasisällöistä edetään yhä enemmän liikkuvaan kuvaan. Myös koko ajan yleistyvä 3D on esitystapana vahvasti tulevaisuutta. Uskoisinkin, että laajennetun todellisuuden matkapuhelinpalvelujen sisällöissä mennään yhä enemmän lähemmäksi realismia. Esimerkiksi vuonna 1999 Oulun yliopistossa toteutetut silmikkonäyttökokeilut<sup>1</sup> ovat osaltaan olleet askeleita tähän suuntaan. Tulevaisuudessa esimerkiksi matkapuhelimen kameran ja suuren näytön avulla palveluiden sisällöt tuodaan käyttäjän katseen ja todellisuuden väliin.

Opinnäytetyössä toteuttamani sovelluspilotit ovat vielä kaukana mainituista tulevaisuudenkuvista. Toteutetut tekniset pilotit ovat kuitenkin tiettävästi ensimmäisiä Pohjois-Karjalassa toteutettuja laajennetun todellisuuden palveluun soveltuvia sovellusaloja. Jatkokehittävää toteutuksissa varmasti vielä on niin sisällöissä kuin teknisissä ratkaisuissa. Ensisijaisena jatkokehitystyönä näkisin palvelualueiden muokkaamisen yhä useamman matkapuhelinvalmistajan laitteille soveltuvaksi. Myös erilaisten sisältömuotojen kuten videokuvan liittäminen mukaan palveluun voisi olla jatkokehityksen kannalta tärkeää. Mikäli palvelu tulotaisiin julkaisemaan, olisi sen käytettävyydessä ja esimerkiksi latausajoissa vielä paljon paranneltavaa. Kuitenkin kun palvelun vertailukohtaksi otetaan vuonna 2010 toteutettu GPS-Taival, on tapahtunut kehitys laitteissa, niiden käytettävyydessä sekä tarjottavissa sisältömuodoissa ollut nopeaa. Mikäli kehitysvauhti jatkuu samansuuntaisena, en pitäisi realismin rikastamista esimerkiksi kolmiulotteisilla näkymillä enää kovinkaan utopistisena kehitysaskeleena.

---

<sup>1</sup> Cyphone-hankkeet ja MARISIL-patentti, <http://www.oulu.fi/yliopisto/>.

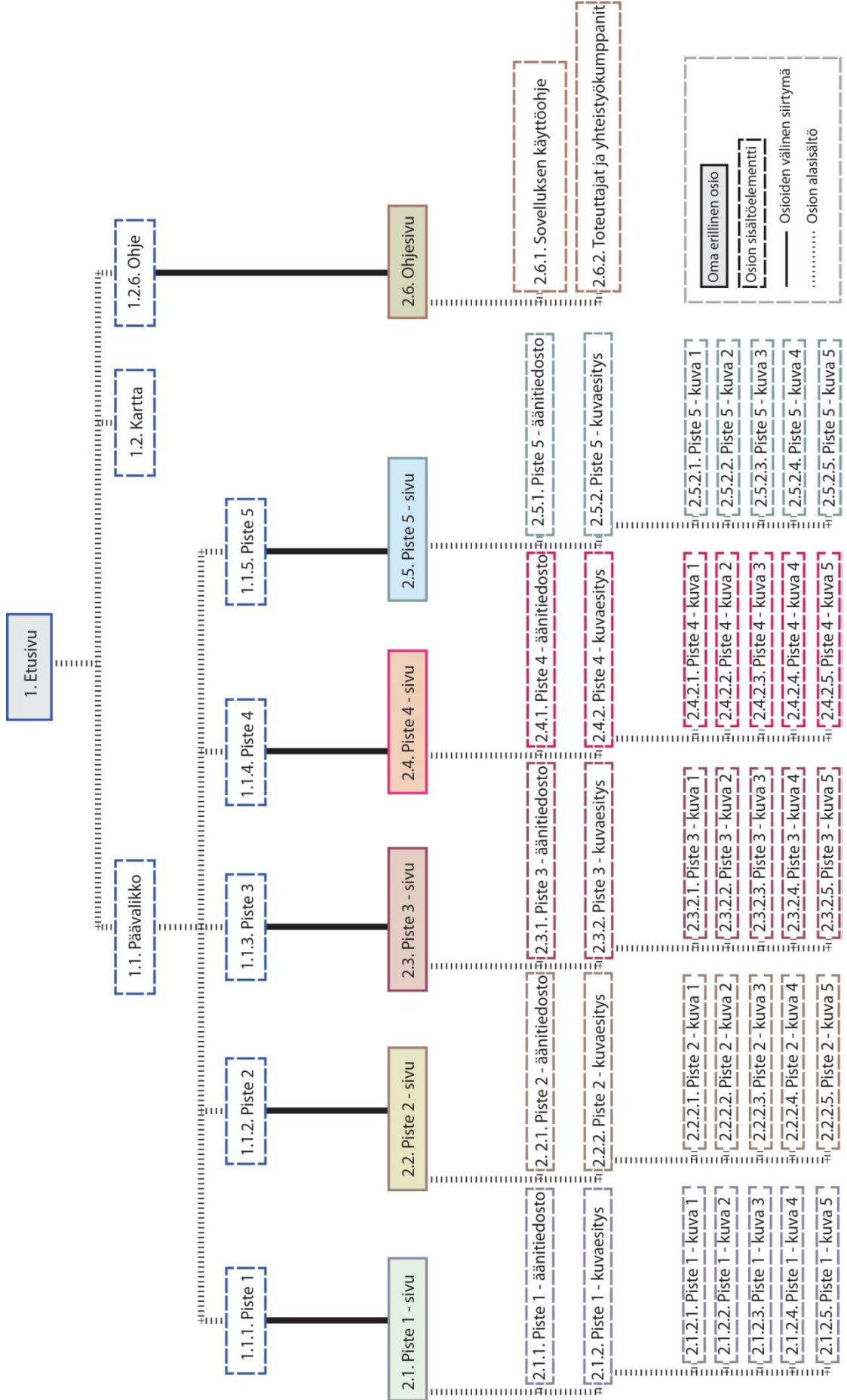
## LÄHTEET

- Blandford R. & Spence E. 2008. The wonderful (widgetised) World of Web Runtime (WRT). All About Symbian.  
[http://www.allaboutsymbian.com/features/item/The\\_Wonderful\\_Widgetised\\_World\\_of\\_Web\\_Runtime.php](http://www.allaboutsymbian.com/features/item/The_Wonderful_Widgetised_World_of_Web_Runtime.php). 11.2.2011.
- Carboni D., Sanna S., Giroux S., Paddeu G. 2011. Interactions Model and Code Generation for J2ME Applications. CRS4.  
[http://www.crs4.it/Publications/cgi-bin/tr/repository/5423\\_0.pdf](http://www.crs4.it/Publications/cgi-bin/tr/repository/5423_0.pdf). 2.3.2011.
- Djuknic G.M. & Richton R.E. 2001. Geolocation and Assisted GPS. Bell Laboratories & Lucent Technologies.  
<http://cens.ucla.edu/~mhr/cs219/location/djunkic01.pdf>. 7.1.2011.
- IBM. 2011. IBM Design: What Is a User Interface?. Design @ IBM.  
<http://www-01.ibm.com/software/ucd/designconcepts/whatisUI.html>. 25.3.2011.
- Java User Group. 2010. MIDP-telephones. Club Java.  
[http://www.club-java.com/TastePhone/J2ME/MIDP\\_mobile.jsp](http://www.club-java.com/TastePhone/J2ME/MIDP_mobile.jsp). 15.10.2010.
- JM-Mobile Solutions. 2008. JM-Mobile User's Guide for the Versions 1.2x.  
[http://www.jm-mobile.com/res/jmm\\_userguide.pdf](http://www.jm-mobile.com/res/jmm_userguide.pdf). 8.10.2010.
- Kowoma. 2009. GPS Explained. <http://www.kowoma.de/en/gps/>. 9.12.2010.
- Kuivakari S. (toim.) 2006. Mobiilikulttuuri. Taiteiden tiedekunnan julkaisuja, katsauksia ja puheenvuoroja 31. Rovaniemi: Lapin yliopisto.
- Mahmoud Q.H. 2002. Deploying Wireless Java Application. Sun Developer Network. <http://developers.sun.com/mobility/midp/articles/deploy/>. 13.1.2011.
- Nitobi. 2011. PhoneGap-documentation. <http://docs.phonegap.com/>. 15.3.2011.
- Nokia. 2010a. Web Runtime Widget. Forum Nokia Library.  
[http://library.forum.nokia.com/index.jsp?topic=/Web\\_Developers\\_Library/GUID-57DAF2D2-28DF-45F4-B901-05F4B2CFF109.html](http://library.forum.nokia.com/index.jsp?topic=/Web_Developers_Library/GUID-57DAF2D2-28DF-45F4-B901-05F4B2CFF109.html). 8.1.2011.
- Nokia. 2010b. Nokia Platform Services 2.0 Release Notes. Forum.Nokia.  
[http://www.forum.nokia.com/dp?uri=http%3A%2F%2Fsw.nokia.com%2Ffid%2F72441975-3f1d-43c0-8e70-f6f360fe2427%2FNokia\\_Platform\\_Services\\_2\\_0\\_Release\\_Notes\\_v1\\_0\\_en.pdf](http://www.forum.nokia.com/dp?uri=http%3A%2F%2Fsw.nokia.com%2Ffid%2F72441975-3f1d-43c0-8e70-f6f360fe2427%2FNokia_Platform_Services_2_0_Release_Notes_v1_0_en.pdf). 5.1.2011.
- Nokia. 2010c. Using Platform Services. Forum Nokia Library.  
[http://library.forum.nokia.com/topic/Web\\_Developers\\_Library/GUID-46EABDC1-37CB-412A-ACAD-1A1A9466BB68.html](http://library.forum.nokia.com/topic/Web_Developers_Library/GUID-46EABDC1-37CB-412A-ACAD-1A1A9466BB68.html). 6.1.2011.
- Nokia. 2011a. Creating a Info.plist-file. Web - Getting Started.  
[http://www.forum.nokia.com/Develop/Web/Getting\\_started/Step\\_3\\_Create.xhtml](http://www.forum.nokia.com/Develop/Web/Getting_started/Step_3_Create.xhtml). 6.1.2011.
- Nokia. 2011b. Device-specification. Forum.Nokia.  
[http://www.forum.nokia.com/Devices/Device\\_specifications/?filter1=webruntime](http://www.forum.nokia.com/Devices/Device_specifications/?filter1=webruntime). 8.3.2011.
- Pihlajamäki T. 2010. Mobiililaitteiden cross-platform ohjelmointi web-tekniikoin. Mobiili ohjelmointi web-tekniikoin.  
<http://teppo.tv/kirjoituksia/mobile.html>. 18.12.2010.

- Ristolainen K. 2007. Matkailu ja kulttuurikohteiden mobiilipalvelut - Soveltuvuus itäsuomalaisiin kulttuurihistoriallisiin kohteisiin. Saimaan laivaston vanavedessä -hankkeen julkaisut ja esitelmät. Joensuun yliopisto. [http://www.uef.fi/c/document\\_library/get\\_file?uuid=49045d4c-db94-4162-b7ed-64d69505ea4e&groupId=180473&p\\_l\\_id=1068110](http://www.uef.fi/c/document_library/get_file?uuid=49045d4c-db94-4162-b7ed-64d69505ea4e&groupId=180473&p_l_id=1068110). 10.11.2010.
- The jQuery Project. 2010. jQuery Mobile: Demos and Documentation. <http://jquerymobile.com/demos/1.0a3/>. 9.2.2011.
- Vesa. J. 2010. Mobiilisovellusten toinen tuleminen. Tietoviikko. Pöhinää-blogi. <http://www.tietoviikko.fi/blogit/pohinaa/article533061.ece>. 7.1.2011.
- Waloszek G. 2000. Interaction Design Guide for Touchscreen Applications. SAP Design Guild. <http://www.sapdesignguild.org/resources/tsdesigngl/TSDesignGL.pdf>. 21.1.2011.
- Ylä-Kotola M. & Arai M. 2000. Uusmediatieteen perusteet. Helsinki: Edita.

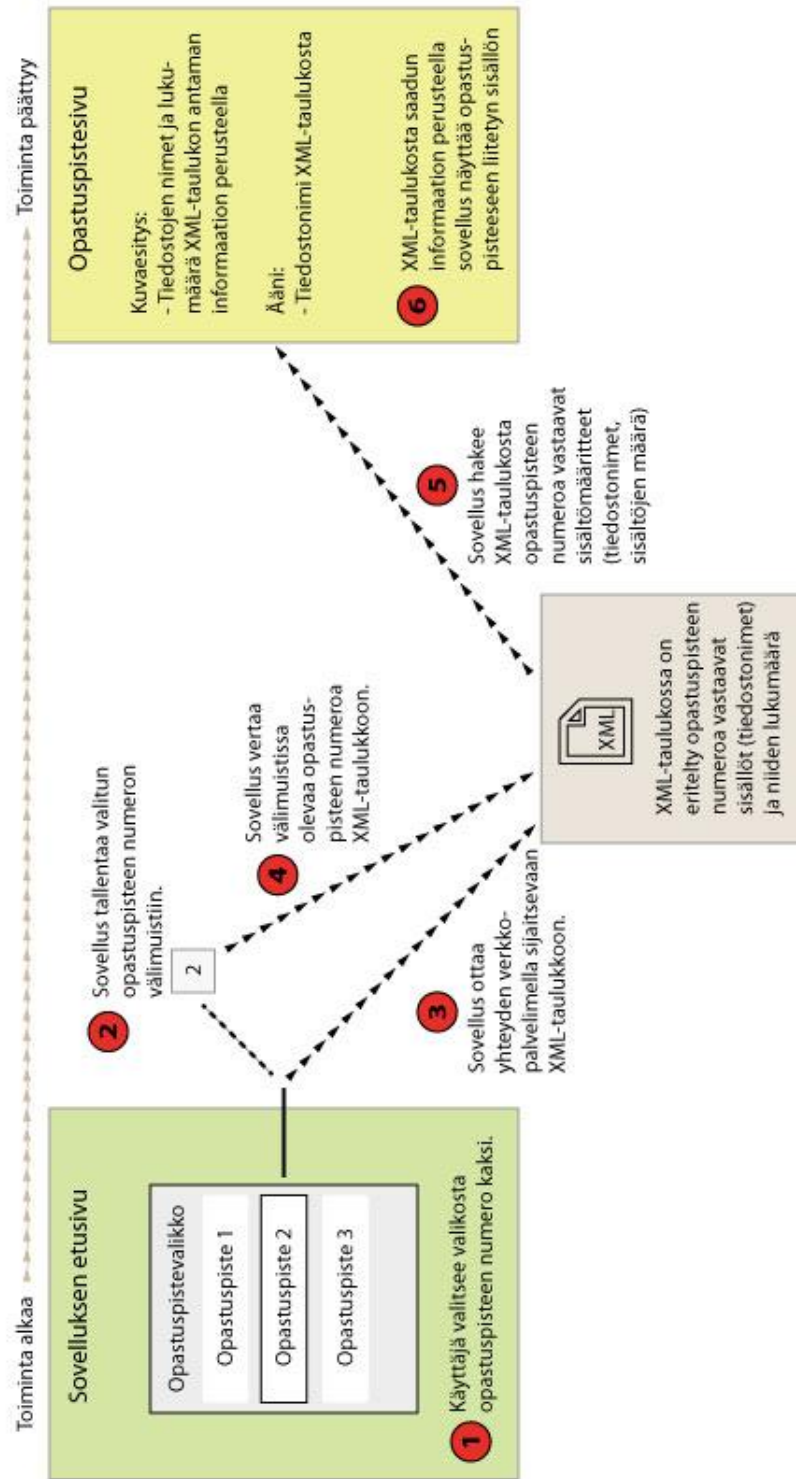


# Vuonislahden kyläkävelyn rakennekaavio



## XML-pohjaisen sovellusrakenteen toimintaperiaate

Vuonislahden Kyläkävely - XML-pohjaisen sovellusrakenteen toimintaperiaate.  
Opinnäytetyö. Tero Hyttinen. Pohjois-Karjalan Ammattikorkeakoulu.

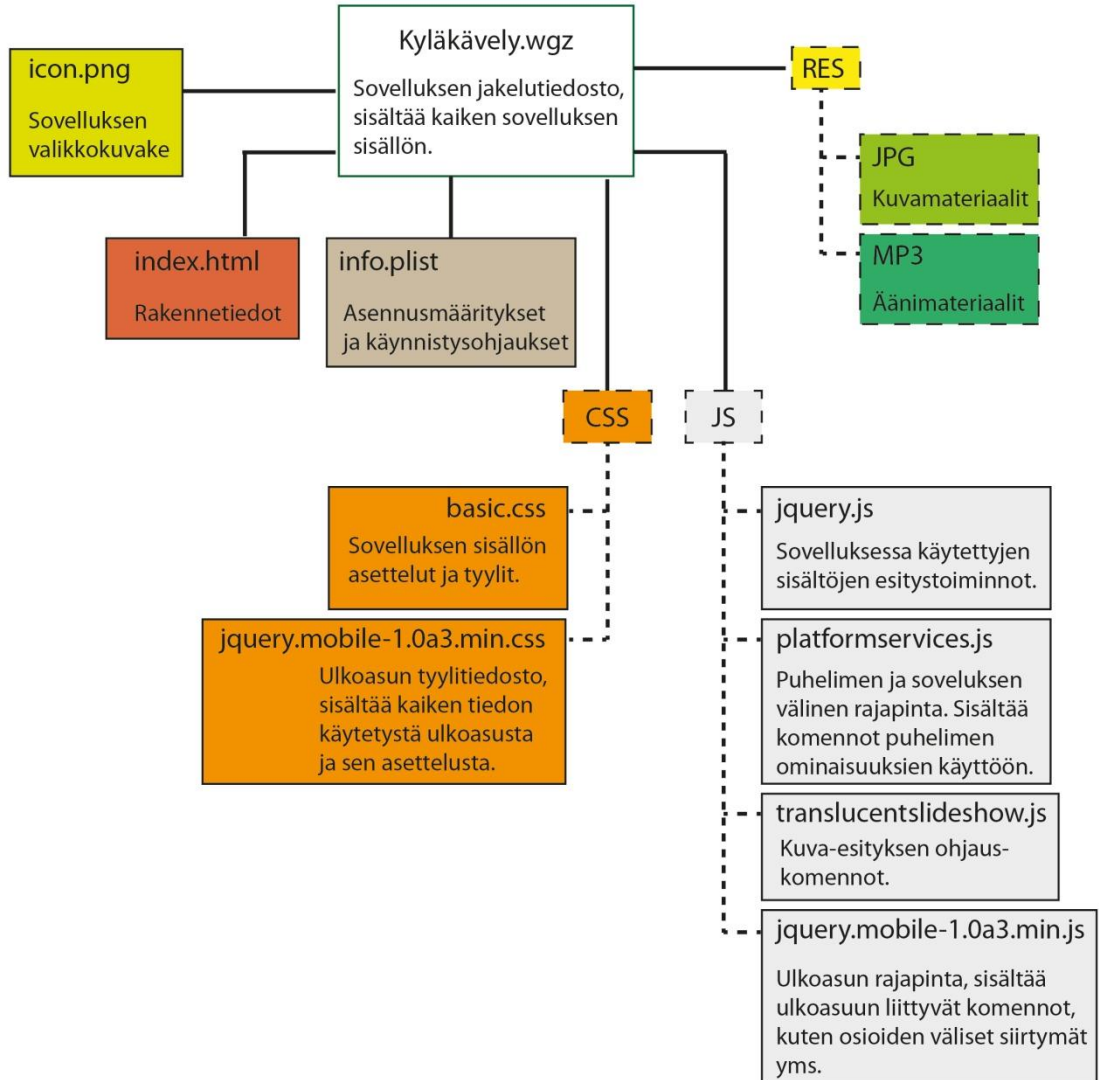


## WRT-sovellusversion kokoonpano

Vuonislahden Kyläkävely - sovelluksen WRT-version kokoonpano

□ - tiedosto

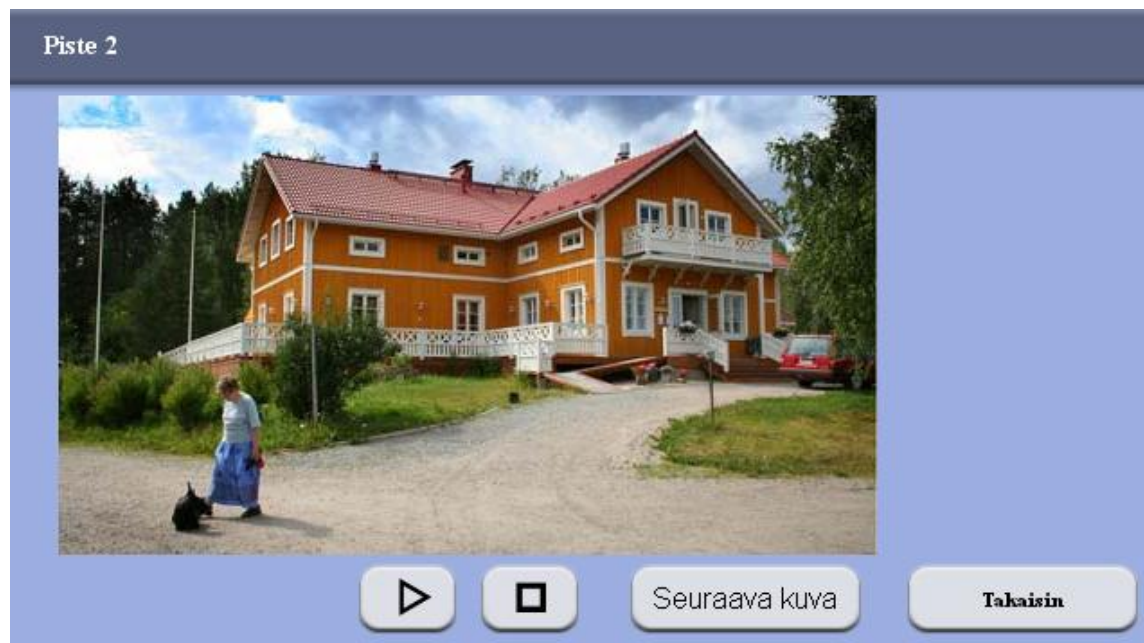
□ - tiedostokansio



## Java ME -sovellusversion ulkoasu



Kuvio 14. Java ME -sovellusversion etusivu.

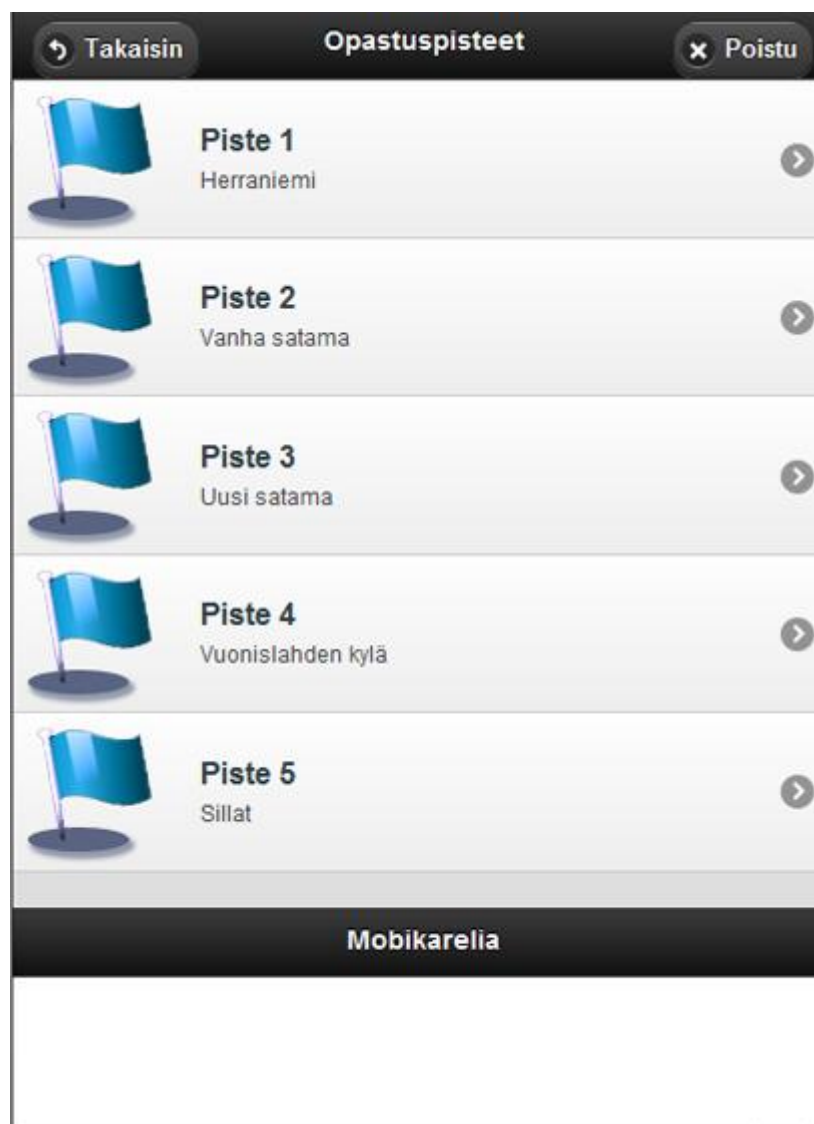


Kuvio 15. Java ME -sovellusversion opastuspisteen sisältösivu.

**WRT-sovellusversion ulkoasu**

Kuvio 16. WRT-sovellusversion etusivu.

## WRT-sovellusversion ulkoasu



Kuvio 17. WRT-sovellusversion opastuspistevalikko.

## WRT-sovellusversion ulkoasu



Kuvio 18. WRT-sovellusversion opastuspisteen sisältösivu.