Teemu Lepola

UPNP OHJAUSPISTE

Opinnäytetyö
KESKI-POHJANMAAN AMMATTIKORKEAKOULU
Tietotekniikka, Toukokuu 2010

KESKI-POHJANMAAN AMMATTIKORKEAKOULU
MELLERSTA ÖSTERBOTTENS YRKESHÖGSKOLA

TIIVISTELMÄ OPINNÄYTETYÖSTÄ

Yksikkö Ylivieska	**Aika** 13.5.2010	**Tekijä** Teemu Lepola
Koulutusohjelma Tietotekniikka		
Työn nimi UPnP ohjauspiste		
Työn ohjaaja Joni Jämsä FM		**Sivumäärä** 38
Työelämäohjaaja		

Opinnäytetyön aiheena oli ohjelmoida Universal Plug and Play-standardin mukainen ohjauspiste mobiililaitteelle. Tavoitteena oli saada ohjauspiste ohjaamaan mediapalvelimia sekä mediatoistimia. Ohjauspisteellä piti pystyä selaamaan mediapalvelimen sisältöä sekä soittaa haluttu kappale mediatoistimessa. Ohjauspisteellä piti pystyä ohjaamaan mediatoistimen toistotilaa.

Ohjelmoinnissa käytettiin Sun microsystemsin kehittämää Java-ohjelmointikieltä sekä Java Micro Edition-sovellusympäristöä. Ohjelmointiympäristönä käytettiin Eclipseä.

Ohjelmoinnin tuloksena ohjauspiste pystyi selaamaan mediapalvelimien sisältöä sekä toistamaan haluttua kappaletta mediatoistimessa. Ohjauspiste pystyi myös muuttamaan mediatoistimen tilaa play, -pause, -sekä stop-tiloihin.

Kokonaisuudessaan työn tavoitteet saavutettiin suurimmalta osalta. Ohjelmoinnissa ei pystytty noudattamaan täysin UPnP:n standardeja Java ME:n vajavaisten ominaisuuksien takia. Työ jäi hyväksi pohjaksi muiden Universal Plug and Play-ohjauspisteiden kehittämiseen.

Asiasanat
Universal Plug and Play, UPnP, JavaME, Mobiiliohjelmointi, Mobiiliohjaus, Eclipse

CENTRAL OSTROBOTHNIA
UNIVERSITY OF APPLIED SCIENCES

ABSTRACT

CENTRAL OSTROBOTHNIA UNIVERSITY OF APPLIED SCIENCES Ylivieska	**Date** 13.5.2010	**Author** Teemu Lepola
Degree programme Information Technology programme		
Name of thesis UPnP control point		
Instructor Joni Jämsä M.Sc		**Pages** 38
Supervisor		

The subject of this thesis was to program Universal Plug and Play standard compatible control point to a mobile phone. The goal was to enable the control point to control media servers and media renderers. The control point had to be able to search for content from media servers and play the desired song in the media renderer. The control point had to be able to control the media renderers rendering state.

Java-programming language and Java Micro Edition platform were used in programming. Eclipse was used as an integrated development environment.

As a result, the control point was able to search for content from media server and play the desired song in the media renderer. The control point was also able to change the rendering state to play, pause or stop state.

As a whole, the goals of the thesis were achieved. It was not possible to fully follow UPnP's standards due to Java ME's lack of features. The thesis serves as a good base for developing other Universal Plug and Play control points.

Key words
Universal Plug and Play, UPnP, JavaME, Mobile programming, Mobile control, Eclipse

LYHENTEET

DHCP	Dynamic Host Configuration Protocol
GENA	General Event Notification Architecture
HTTP	Hyper Text Transport Protocol
IP	Internet Protocol
Java ME	Java Micro Edition
Java SE	Java Standard Edition
RPC	Remote Procedure Call
SOAP	Simple Object Access Protocol
SSDP	Simple Service Discovery Protocol
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
UPnP	Universal Plug and PLay
UPnP/AV	Universal Plug and Play Audio & Video
URI	Uniform Resource Identifier
USN	Unique Service Name
UUID	Universally Unique Identifier
XML	Extensible Markup Language

1 JOHDANTO

Jokaisessa kotitaloudessa on suuri määrä erilaisia elektroniikkatuotteita viihdelaitteista aina hyötylaitteisiin. Laitteina voi olla esimerkiksi televisio, tietokone, pyykinpesukone ja paljon muuta. Kuitenkin kaikilla näillä laitteilla on yksi yhteinen ominaisuus – laitteet toimivat itsenäisesti eivätkä ne kommunikoi toistensa kanssa. Universal Plug and Play (UPnP) on joukko verkkoprotokollia, joilla mahdollistetaan eri verkkolaitteet toimimaan yhdessä. Laitteita voivat olla esimerkiksi kännykkä, stereot tai tietokone. UPnP mahdollistaa kännykän ohjaamaan stereoita soittamaan musiikkikappaletta, joka sijaitsee tietokoneella. UPnP on ollut jo pitkään markkinoilla, mutta vasta viime vuosina se on levinnyt laajemmalti moniin elektroniikkalaitteisiin. UPnP:n käyttökohteet eivät pelkästään rajoitu viihdelaitteisiin. Käytännössä UPnP:llä voi ohjata mitä tahansa hehkulampusta koko huoneiston lämmitysjärjestelmiin ja ilmastointiin.

Opinnäytetyön aiheena on UPnP control pointin eli ohjauspisteen ohjelmoiminen käyttäen Java Micro Editionia. Ohjauspisteen tarkoitus on ohjata mediasoittimia ja mediapalvelimia. Opinnäytetyön tavoitteena on ohjelmoida ohjauspiste, jolla voi vähintäänkin selata mediapalvelimien sisältöä ja soittaa haluttua mediakohdetta. Lisäksi tavoitteena on saada kokonaisvaltainen ymmärrys UPnP-teknologiaan. Työn tutkittava alue keskittyy UPnP teknologian ja control pointin ympärille.

Työn tekstiosuudessa aluksi tutustutaan UPnP-standardiin. Standardista esitellään UPnP:n eri komponentit ja standardin tärkeimmät protokollat. Jokaisesta protokollasta annetaan yleisellä tasolla kuvaus, miten ne toimivat ja mitä tekniikkaa kukin niistä käyttää. Tekstiosuuden loppu koostuu itse ohjelmointityöstä. Tekstissä kuvataan ohjelmointityön vaiheet sekä taustaa ohjelmointityöstä. Lopuksi on vielä pohdintaa ohjelmointityöstä sekä muuta pohdintaa aiheeseen liittyen.

2 UNIVERSAL PLUG AND PLAY

Tässä luvussa tutustutaan Universal Plug and Play-teknologiaan. Luvussa kerrotaan aluksi UPnP:n standardista sekä yrityksistä UPnP:n takana. Sen jälkeen esitellään UPnP-verkon komponentit sekä tehdään tarkka katsaus UPnP:n protokolliin. Lopuksi tutustutaan Universal Plug and Play Audio & Videoon.

2.1 Standardi

Universal Plug and Play on joukko verkkoprotokollia, joilla mahdollistetaan eri verkkolaitteet toimimaan yhdessä. Protokollat yhdessä määrittelevät UPnP standardin. Protokollien tarkoitus on helpottaa laitteiden yhdistämistä ja edistää laitteiden välistä kommunikaatiota. Kommunikaatio voi olla esimerkiksi viestien lähettämistä tai tiedostojen jakamista laitteiden kesken. Laitteita voi olla esimerkiksi Personal Computer (PC), mobiililaite, televisio tai jokin muu kodin elektroniikkalaite. UPnP:n määrittelemät protokollat pohjautuvat jo aiempiin Internet-pohjaisiin verkkoprotokolliin, kuten Transmission Control Protol (TCP), Internet Protocol (IP), Hyper Text Transport Protocol (HTTP) sekä Extensible Markup Language (XML). Kaikki UPnP-teknologiaa käyttävät laitteet tulee toteuttaa protokollien määrittämät vaatimukset. Vaatimuksina on mm. tarjota palvelut laitteen löytämiseen, ohjaamiseen sekä tiedonsiirtoon UPnP-laitteiden välillä. UPnP:n käyttämien yleisien ja tunnettujen protokollien ansiosta UPnP-teknologia voidaan tuoda mille tahansa tunnetulle alustalle, riippumatta käyttöjärjestelmästä ja riippumatta siitä, tapahtuuko tiedonsiirto langallisesti vai langattomasti. (Jeronimo & Weast 2003, 5; Wikipedia 2010; UPnP Forum 2008.)

UPnP-arkkitehtuuriin on rakennettu useita erilaisia ominaisuuksia. Näitä ominaisuuksia on mm. laitteiden liitettävyys, Ad-hoc-verkot, nolla-konfiguraatio-yhteydet, standardipohjainen arkkitehtuuri, alustariippumattomuus, laite- ja

väyläriippumattomuus sekä ohjelmallinen ja manuaalinen ohjaus. (Jeronimo & Weast 2003, 6; Wikipedia 2010.)

Laitteiden liitettävyys. UPnP-arkkitehtuuri määrittelee laitteille protokollat, joilla laitteet voivat keskustella toistensa kanssa. Laitteet voivat liittyä ja poistua verkosta ilman sen suurempia toimenpiteitä. Laitteet voivat myös mainostaa omia palvelujaan, etsiä ja löytää toisia laitteita, lähettää laitteille viestejä tai ohjata jotain toista laitetta (Jeronimo, Weast 2003, 5).

Nolla-konfiguraatio ja Ad-hoc verkot. UPnP-arkkitehtuuri on rakennettu tukemaan nolla-konfiguraatiota. Nolla-konfiguraatio tarkoittaa sitä, että loppukäyttäjän ei tarvitse tehdä mitään säätöjä ennen laitteen käyttöönottoa. Nolla-konfiguraatio on joukko tekniikoita, jotka automaattisesti rakentavat käytettävän IP-verkon. Tällaisia verkkoja kutsutaan Ad-hoc-verkoiksi. Verkon muodostuksessa ei tarvita mitään valmiina olevaa verkkoinfrastruktuuria, kuten reitittimiä tai muita palvelimia. (Jeronimo & Weast 2003, 6, Wikipedia 2010.)

Standardipohjainen arkkitehtuuri. Koko UPnP-arkkitehtuuri on rakennettu valmiiden ja avoimien standardien pohjalle. Standardeja on mm. IP, TCP, HTTP, XML yms. Valmiiden standardien käyttäminen helpottaa UPnP-laitteiden kehittämistä (Jeronimo, Weast 2003, 6).

Alustariippumattomuus. UPnP on pelkästään joukko protokollia eikä esimerkiksi ohjelmointirajapinta. Protokollat on suunniteltu niin, että protokolla toimii riippumatta alustasta tai käyttöjärjestelmästä. Tällöin laitevalmistajat eivät joudu itse rakentamaan toteutusta jollekin tietylle alustalle, vaan yksi ratkaisu toimii jokaisella alustalla ja eri valmistajien tekemät laitteet ovat keskenään yhteensopivia (Jeronimo & Weast 2003, 6; Wikipedia 2010).

Laite- ja väyläriippumattomuus. UPnP teknologia toimii missä tahansa tietoliikenneväylässä, missä liikkuu IP-pohjainen liikenne. Tällaisia väyliä on mm.

puhelin – ja voimalinjat, Ethernet, FireWire, BlueTooth ja Wi-Fi. Erillisiä laiteajureita ei tarvita, vaan UPnP käyttää yleisesti tunnettuja protokollia (Jeronimo & Weast 2003, 6).

Ohjelmallinen ja manuaalinen laitteiden ohjaus. UPnP-laitteita voidaan ohjata ohjelmallisesti tai manuaalisesti laitteen selainpohjaisella käyttöliittymällä (Jeronimo & Weast 2003, 6; Wikipedia 2010).

2.2 Yritykset UPnP:n takana

UPnP:tä kehittää UPnP Forum. UPnP Forum on joukko yrityksiä, jotka määrittelevät UPnP:n liittyviä laite- ja palvelukuvauksia. UPnP Forum perustettiin vuonna 1999 ja nykyään Foorumiin kuuluu lähes 900 yritystä tai muuta tahoa, jotka kehittävät UPnP:tä. Foorumin tavoitteina on mahdollistaa helposti verkkoon yhdistettävien laitteiden kehittäminen ja yksinkertaistaa verkkojen toteutuksia koti- ja yritysympäristöissä. Foorumin Internet-sivu, http://www.upnp.org, on foorumin keskuspaikka, josta voi seurata sen uusimpia uutisia ja lukea dokumentteja. (UPnP Forum 2000, 2010.)

2.3 UPnP-verkon komponentit

Tässä luvussa esitellään UPnP-verkon yleisimmät komponentit. Komponenteista kerrotaan lyhyesti niiden ominaisuuksista sekä roolista UPnP-verkoissa. Yleisimmät UPnP-verkon komponentit ovat device, service sekä control point.

2.3.1 Device

Device, laite, on UPnP-verkossa laite, joka sisältää palveluita ja mahdollisesti sisäisiä laitteita. Laite voi olla käytännössä mikä tahansa kodin elektroniikkalaite; esimerkiksi PC, videonauhuri, stereot, pesukone tai valokatkaisin. Laitteena voisi olla myös televisio, jonka sisällä on videonauhuri. Tällöin videonauhuri on television sisäinen laite. Jokaiselle laitteella on laitekuvaus, jossa on tiedot laitteen sisäisistä laitteista ja ennen kaikkea laitteen palveluista. Laitekuvaus on XML-tiedostona. Jokaisella laitteella on erilaisia palveluita. Esimerkiksi stereolla voisi olla palvelu, jonka kautta voidaan toistaa musiikkia, pysäyttää toisto tai vaikka nauhoittaa musiikkia. Laitekuvauksessa on myös yleisiä laitteen tietoja, kuten nimi, kehittäjä yms. (Jeronimo & Weast 2003, 25; UPnP Forum 2000.)

2.3.2 Service

Serviceä, palvelua, voidaan pitää hyvin samanlaisena kuin jonkin olio-ohjelmointikielen liittymää. Palvelu on kokonaisuus, joka pitää sisällään tietoa ja toiminnallisuutta. Toiminnallisuudella tarkoitetaan palvelun toimintoja. Toiminnoilla voidaan ohjata palvelua. Palvelulla voi olla useita toimintoja mutta, niitä voi olla myös nolla. Jokaisella toiminnolla on yksi tai useampi argumentti. Argumentit ilmaisevat palvelun tilaa. Jokaisella argumentilla on nimi, arvo ja suunta. Suunta on joko sisään tai ulos, mutta ei molempia. Sisään tuleva argumentti vaikuttaa palvelun sisäiseen tilaan ja ulospäin menevä argumentti lähetetään eteenpäin sille taholle, joka laukaisi ko. toiminnon. Kello-laitteella voisi olla esimeriksi palvelu, jolla on kaksi toimintoa. Toiminnot ovat aseta aika ja näytä aika. Toiminnoilla on argumentti aika, joka sisältää tämän hetken kellonajan. Näytä aika-toiminnolla kello näyttää kellonajan ja aseta aika-toiminnolla kelloon voi asettaa uuden ajan. Tiedot palveluiden toiminnoista ja argumenteista on XML-dokumentissa. Laitteen laitekuvauksessa on linkki palveluiden XML-dokumentteihin. (Jeronimo & Weast 2003, 25; UPnP Forum 2000.)

Jokaisella palvelulla on argumenttitaulu, ohjaus- ja ilmoituspalvelin. Argumenttitaulu sisältää kaikki palvelun argumentit. Ohjauspalvelin ottaa vastaan toimintopyyntöjä, suorittaa toimintoja, päivittää argumenttitaulun sekä palauttaa vastauksia toimintoja pyytäneille ohjauspisteille. Ilmoituspalvelin toimii nimensä mukaisesti ilmoittajana. Ilmoituspalvelin lähettää tapahtumia niistä kiinnostuneille laitteille. Ilmoitusten tarkoitus on kertoa muille laitteille laitteen tilasta. Ilmoitukset lähetetään aina, kun jonkun argumentin arvoa muutetaan tai jokin toiminto suoritetaan. Ilmoitus lähetetään niistä kiinnostuneille kuuntelijoille, jotka ovat rekisteröityneet kuuntelemaan laitetta. Esimeriksi palohälytin voisi lähettää viestin, kun se havaitsee savua. (Jeronimo & Weast 2003, 25; UPnP Forum 2000.)

2.3.3 Control Point

Control Point, ohjauspiste, toimii nimensä mukaisesti pisteenä, joka voi etsiä ja ohjata muita laitteita. Kun ohjauspiste on etsinyt ja löytänyt jonkin ohjattavan laitteen, piste voi pyytää listaa laitteen palveluista, pyytää tarkempia tietoja jostain palvelusta, lähettää käskyjä laitteelle sekä rekisteröityä laitteelle kuunnellakseen laitteen lähettämiä ilmoituksia. (Jeronimo & Weast 2003. 18; UPnP Forum 2000.)

2.4 Protokollat

Tässä luvussa esitellään UPnP:n protokollat. Protokollat määrittelevät eri vaiheet, mitä laitteet ja ohjauspisteet tekevät kytkeytyessään verkkoon. Protokollat määrittelevät, miten UPnP laitteet sekä ohjauspisteet ylipäänsä toimivat. Protokollat ovat addressing, discovery, description. control, eventing sekä presentation.

2.4.1 Addressing

Ensimmäinen vaihe, kun laite ottaa yhteyden paikalliseen verkkoon, on IP-osoitteen saaminen laitteelle. Jokainen UPnP laite erotetaan toisistaan IP-osoitteiden avulla. Laitteet käyttävät myös IP-osoitetta laitteiden väliseen kommunikaatioon, sillä ilman IP-osoitetta laitteet eivät tietäisi, mille laitteelle viesti tulisi lähettää. (Jeronimo & Weast 2003, 65; Wikipedia 2010; UpnP Forum 2000.)

UPnP-arkkitehtuuri esittelee kaksi eri tapaa hankkia IP-osoite: Dynamic Host Configuration Protocol (DHCP) ja Auto-IP. DHCP on yleisesti käytössä oleva järjestelmä, jossa käytetään erillistä palvelinta, joka hallitsee verkon IP-osoitteita. Auto-IP on järjestelmä, jota käytetään silloin, kun DHCP:tä ei ole saatavilla. Järjestelmän idea on se, että verkon laitteet toimivat itse IP:n hankkimisessa. Molempien järjestelmien tarkoitus on helpottaa loppukäyttäjää. Molemmat tavat toimivat automaattisesti, eikä loppukäyttäjän tarvitse itse säätää yhteysasetuksia. (Jeronimo & Weast 2003, 65-74; Wikipedia 2010; UPnP Forum 2000.)

IP-osoitteen hankkimisessa DHCP:tä käytetään ensijaisesti. UPnP-laitteen ottaessa yheyden verkkoon se hakee ensimmäiseksi DHCP-palvelinta verkosta. Palvelimen löydyttyä laite pyytää palvelimelta IP-osoitetta. Palvelin lähettää laitteelle ehdotuksen IP-osoitteesta ja laite hyväksyy ehdotuksen. Näin laitteella on nyt IP-osoite, jonka jälkeen voidaan alkaa tekemään seuraavia toimenpiteitä. DHCP-palvelimella on kolme tapaa lähettää IP-osoite. Yleisimmässä tavassa IP-osoite annetaan käyttöön ainoastaan tietyn mittaiselle aikajaksolle. Kun aikajakso on mennyt ohi, laitteen tulee pyytää uudestaan IP-osoitetta. DHCP:ssä on järjestelmä, joka pitää kirjaa vanhentuneista IP-osoitteista. Tämä takaa sen, että laite, jonka IP-osoite on vanhentunut, saa pyytäessään käyttöönsä aikaisemmin käytössä olleen osoitteensa. Tällöin laitteiden osoitteet eivät vaihdu jatkuvasti, vaan pysyvät koko ajan samana. (Jeronimo & Weast 2003, 65-69; Wikipedia 2010; UPnP Forum 2000.)

Kun DHCP-palvelinta ei ole saatavilla verkossa, laitteiden on käytettävä Auto-IP järjestelmää. Järjestelmässä laite valitsee itselleen IP-osoitteen ja kyselee muilta laitteilta, onko valittu IP-osoite jo käytössä. Jos IP on jo käytössä, laite valitsee uuden ja kysyy uudestaan. Tätä jatketaan niin kauan, että löytyy IP-osoite, joka ei ole käytössä. Valittava IP-osoite on IP-alueella 169.254/16. Tätä osoiteväliä ei reititetä ulkomaailmaan, esimerkiksi Internetiin. Laitteet pysyvät omana sisäisenä verkkona. Osoiteväli on kansainvälisesti sovittu ja on tarkoituksellisesti pidetty alueena, jota ei reititä. Tätä osoiteväliä kutsutaan myös nimellä LinkLocal. Osoitevälin osoitemaski on b-luokan maski 255.255.0.0. Auto-IP järjestelmän ansiosta laitteet voivat muodostaa Ad-hoc verkkoja. Verkko luodaan kokonaan laitteiden toimesta ilman muiden osapuolien, kuten DHCP-palvelimien, ottaessa osaa verkon rakentamiseen. Auto-IP:llä on kuitenkin rajoituksia. Laitteet eivät pääse koskaan ulkomaailmaan, jolloin laitteet pysyvät yhtenä sisäisenä verkkona. Lisäksi Auto-IP:llä verkon rakentaminen kestää ajallisesti selvästi kauemmin kuin DHCP:llä. Auto-IP on pelkästään toissijainen ratkaisu ja UPnP-laite tarkistaa aina tietyin väliajoin, onko DHCP-palvelinta olemassa. Jos palvelin löytyy, UPnP-laite siirtyy heti käyttämään palvelinta ja unohtaa Auto-IP:llä hankkiman osoitteen. (Jeronimo & Weast 2003, 65-69; Wikipedia 2010; UPnP Forum 2000.)

2.4.2 Discovery

UPnP-laitteen saatua IP-osoitteen, on sen vuoro antaa tietoa ohjauspisteille itsestään. Discoveryn yleinen tarkoitus on se, että verkon ohjauspisteet etsivät ja löytävät verkossa olevia laitteita sekä palveluita. Laitteiden tulee mainostaa itseään sekä laitteen palveluita. Ohjauspisteiden tulisi tietää jokaisesta verkossa olevasta laitteesta ja palvelusta, josta se on kiinnostunut. (Jeronimo & Weast 2003, 75-76; Wikipedia 2010; UPnP Forum 2000.)

Simple Service Discovery Protocol (SSDP) on protokolla, jota UPnP käyttää hyväkseen discoveryssä. SSDP tuo ratkaisun laitteiden etsimiseen, löytämiseen ja

mainostamiseen. Protokolla on rakennettu HTTP-pohjaisen tiedon jakamiseen verkossa. SSDP tuo UPnP-laitteille uusia ominaisuuksia. Ominaisuudet ovat palvelutyyppi sekä Unique Service Name (USN). Palvelutyyppi on Uniform Resource Identifier (URI), joka ilmaisee tyypin jollekin palvelulle. USN on URI, joka identifioi jokaisen palvelun. USN sisältää Universally Unique Identifier (UUID)-luvun, joka on 128-bitin numerosarja. Numerosarjalla erotetaan eri palvelut toisistaan. UPnP:tä kehittävät tahot ovat määrittäneet selvät laite –ja palvelutyypit kullekin standardille laitetyypille. (Jeronimo & Weast 2003, 78-79.)

SSDP käyttää liikennöimiseen hajautettua mallia, jossa laitteet keskustelevat suoraan toisten laitteiden kanssa. SSDP käyttää lähetysten sovelluskerroksena HTTP:tä ja lähetyskerroksena on User Datagram Protocol (UDP). Jokainen discovery-vaiheen viesti lähetetään UDP:ta käyttäen multicast-osoitteeseen, joka on 239.255.255.250:1900. Kyseinen osoite on SSDP:n varattu multicast-osoite. Viestien data kulkee HTTP:n mukana HTTP-headereissa. (Jeronimo & Weast 2003, 78-80; UPnP Forum 2000.)

SSDP esittelee kahdenlaisia viestejä, joita voidaan lähettää: hakuviestejä ja olemassaoloviestejä. Ohjauspisteet lähettävät hakuviestejä. Hakuviestien tarkoitus on saada ohjauspisteen tietoon verkossa olevista laitteista ja niiden palveluista. Viesti on kaksiosainen lähetys/vastaus-mallia, jossa ohjauspiste lähettää viestin kaikille laitteille ja jokainen laite vastaa ohjauspisteelle. Laitteet lähettävät olemassaoloviestejä. Olemassaoloviestejä on kahdenlaisia. Viesti, kun laite ottaa yhteyden verkkoon ja viesti, kun laite poistuu verkosta. Molempien viestien tarkoitus on informoida ohjauspisteitä. Laitteen tullessa verkkoon, se lähettää olemassaoloviestin kaikille ohjauspisteille. Viesti sisältää tiedon laitteet palveluista ja mahdollisista sisäisistä laitteista. Viestillä siis mainostetaan laitteen palveluita. Kun laite poistuu verkosta, se lähettää poistumisviestin, jolloin ohjauspisteille informoidaan, etteivät laitteen palvelut ole enää käytettävissä. Kuten aiemmin mainittiin, kaikkien viestien data kulkee HTTP:n headereissa. Headerien sisältö riippuu viestistä, mutta jokaisessa viestissä on muutamia peruselementtejä, jotka

löytyvät jokaisesta viestistä. Esim. tieto, miltä laitteelta lähetetty viesti on peräisin. (Jeronimo & Weast 2003, 80–91; UPnP Forum 2000, Wikipedia 2010.)

2.4.3 Description

Ohjauspisteen saatua tietoonsa kaikki verkon laitteet discovery-vaiheessa, on sillä vielä varsin vähän tietoa laitteista. Käytännössä ohjauspiste tietää laitteista vain laitetyypin, sen USN:n sekä linkin laitteiden laitekuvauksiin. Laitekuvaus on hyvin keskeisessä roolissa description-vaiheessa. Tässä vaiheessa ohjauspisteet kyselevät tarkempia tietoja laitteilta. Haluttuja tietoja on mm. mitä palveluita laitteella on. Jokaisesta palvelusta on myös saatavilla tarkemmat tiedot. (Jeronimo & Weast 2003, 93; UPnP Forum 2000.)

Laitteiden ja palveluiden tiedot on tallennettu laite- ja palvelukuvauksiin. Kuvaukset ovat XML-dokumentteja, joissa XML-tagit ovat tietotyyppejä. Niiden sisällä on tyypin arvo. Laitekuvauksissa on listattuna yleisiä tietoja laitteesta, kuten laitteen tyyppi, laitteen nimi, valmistaja, linkki valmistajan kotisivuille, malli, mallin linkki, mallinumero yms. Laitekuvauksissa on myös lista laitteen palveluista. Laitekuvauksessa palveluista saa tietoon mm. palvelutyypin, palvelun nimen sekä linkit palvelun tarkempiin palvelukuvauksiin, laitteen ohjaukseen sekä ilmoituksiin. Palvelukuvauksessa on tarkemmat tiedot palvelusta. Kuvauksessa on tieto palvelun versionumerosta, lista palveluun liittyvistä toiminnoista ja toimintojen argumenteista. (Jeronimo & Weast 2003, 95-107; UPnP Forum 2000; Wikipedia 2010.)

Laite- ja palvelukuvausten lähettäminen tehdään HTTP-protokollaa käyttäen. Ohjauspiste pyytää kuvauksia tekemällä HTTP GET-pyynnön. Pyynnössä on linkki halutun laitteen laitekuvauksesta. Laitteet lähettävät samaa protokollaa käyttäen vastauksen ohjauspisteille. Laitteen ja palvelun XML-kuvaukset lähetetään vastauksen mukana. (Jeronimo & Weast 2003, 108-109.)

2.4.4 Control

Ohjauspisteen saatua tiedot laitteista ja palveluista, ohjauspiste voi ruveta suorittamaan toimintoja eri laitteille. Ohjauspiste voi siis ohjata laitteiden toimintaa. Tästä vaiheesta UPnP:n protokollissa käytetään nimitystä control. Toimintojen laukaisemiseen ohjauspisteen ja laitteen välillä käytetään protokollaa nimeltä Simple Object Access Protocol, (SOAP). SOAP on protokolla, joka mahdollistaa web-pohjaisen viestien lähettämisen. SOAP kytkee sisäänsä Remote Procedure Call (RCP)-protokollan. RCP on protokolla, jolla voidaan laukaista jokin toiminta jossain laitteessa. UPnP:ssä ohjauspiste lähettää SOAP-viestin laitteelle, jossa toiminta laukaistaan. (Jeronimo & Weast 2003, 111-112; Wikipedia 2010; UPnP Forum 2000.)

SOAP-viestien data kulkee XML-dokumentissa ja viestien lähettämiseen käytetään HTTP:tä. SOAP-viestit koostuvat UPnP:ssä neljästä eri osasta. Kaksi näistä osista on XML-skeemoja. Skeemat määrittelevät XML-dokumentin rakenteen, sen säännöt ja tietotyyppien esittelyt. Lisäksi skeemoissa on tieto, mitä lähetystapaa käytetään viestien lähettämisessä. Vaihtoehtoja on useita, mutta pääasiallisesti SOAP:ia käytetään HTTP:n yli. Viimeinen osa koostuu sopimuksesta, miten RCP – kutsut lähetetään ja miten ne vastaanotetaan. UPnP:ssä SOAP-viestit on yleisesti sisällytetty ohjelmointirajapintojen sisään, eikä ohjelmoijan tarvitse niihin kajota kovinkaan syvällisesti. (Jeronimo & Weast 2003, 111-116.)

Laitteiden ohjaus toimii siten, että ohjauspiste lähettää toimintopyyntöviestin halutun laitteen palvelulle. Viesti on käytännössä SOAP-viesti. Viesti sisältää XML:ssä dataa, kuten toiminnan ja sen argumentit. Laitteen saatua viestin, se lukee XML-datasta, mitä ohjauspiste haluaa laukaista ja toimii sen mukaan. Laite lähettää sen jälkeen vastauksen, joka on hyvin samanlainen SOAP-viesti. (Jeronimo & Weast 2003, 113; UPnP Forum 2000; Wikipedia.)

Otetaan esimerkki jo ennestään tutusta kellosta. Kello on laite, jolla on laitekuvaus. Laitekuvauksesta käy ilmi, että kellolla on palvelu nimeltä nykyinen aika. Nykyinen aika-palvelulla on kaksi toimintoa, aseta aika ja näytä aika. Aseta aika-toiminnolla on yksi argumentti nimeltä uusi aika. Argumentin suunta on sisään, jolloin argumentti vaikuttaa palvelun tilaan. Kun aseta aika-toiminto suoritetaan, kelloon asetetaan uusi aika, joka on argumentin mukana tuleva aika. Toinen toiminto, näytä aika, sisältää argumentin nykyinen aika ja argumentin suunta on ulos. Tämä argumentti ei siis vaikuta kellon sisäiseen tilaan, vaan antaa tietoa siitä muille osapuolille. Kun näytä aika-toiminto suoritetaan, palvelu lähettää viestin sitä kutsuneelle osapuolelle ja tietona lähetetään kellon näyttämä aika.

2.4.5 Eventing

UPnP-arkkitehtuurissa ohjauspisteet voidaan asettaa kuuntelemaan jotain laitetta tai palvelua sen tilamuutoksista. Kun laitteen jokin tila muuttuu, se lähettää ilmoituksen ohjauspisteille. Ohjauspisteiden tulee rekisteröityä kuuntelemaan laitteita, jotta ne saisivat ilmoitusviestejä laitteilta. UPnP käyttää ilmoituksissa julkaisija/tilaaja-mallia. Ohjauspisteet ikään kuin tilaavat eli rekisteröityvät kuuntelemaan laitteita ja laitteet julkaisevat tiedotuksia, kun laitteiden sisäinen tila muuttuu. (Jeronimo & Weast 2003, 129-130; Wikipedia 2010, UPnP Forum 2008.)

UPnP käyttää hyväkseen General Event Notification Architecture (GENA)-arkkitehtuuria. GENA on malli, jossa tilaajat voivat pyytää, uusia tai perua tilauksia. Jokaiselle hyväksytylle tilaukselle annetaan tilausnumero. Tilausnumerolla erotetaan tilaukset toisistaan. Numeroa käytetään hyväksi mm. pyynnöissä, joissa tilaus halutaan uusia tai perua. Jokaisella tilauksella on aika, kuinka kauan tilaus on voimassa. Ajan mennessä umpeen, tilaajan on uusittava tilaus. GENA käyttää HTTP:tä tilaus- ja julkaisuviestien lähettämiseen. Viestien data kulkee HTTP-headereissa. GENA esittelee kolme headeria: subscribe, jota käytetään, kun laitteelta tilataan julkaisuja, unsubscribe, kun tilaus peruutetaan ja notify, joka on laitteen

lähettämä julkaisu. Käytössä on myös joukko muita headereita: CALLBACK, NT, NTS ja SID. CALLBACK-headeria käytetään tilaajan ja julkaisijan kommunikaatiossa, tilaaja lähettää tilauksessa tämän headerin, joka sisältää linkin tilaajan osoitteeseen. Laite käyttää julkaisuissa sen jälkeen tätä tietoa, jolloin julkaisu menee julkaisun tilanneelle ohjauspisteelle. NT kertoo julkaisun tyypin. NTS on NT:n alityyppi, joka siis kertoo hieman tarkemmin, minkä tyyppinen julkaisu on. SID on tilausnumero. GENA käyttää myös joukon muuten yleisesti HTTP:ssä käytössä olevia headereita. (Jeronimo & Weast 2003, 130-132; Wikipedia 2010, UPnP Forum 2008.)

2.4.6 Presentation

UPnP-laitteet käyttävät webbiin liittyviä protokollia niin paljon, että voidaan sanoa, että jokaisessa UPnP-laitteessa on sisään rakennettu web-palvelin. Palvelimen avulla UPnP-laitteille voidaan rakentaa web-käyttöliittymä. Tämä mahdollisuus on yksi UPnP:n protokollista, nimeltään presentation. Käyttöliittymän avulla laitteen ominaisuuksia voidaan muuttaa, suorittaa toimintoja laitteille tai tarkastella laitteen tilaa. Web-käyttöliittymä on siis normaali web-sivu. Kuten muiden web-sivujen tapaan, myös tällä on sivulla osoite. Osoite on saatavilla laitteiden laitekuvauksissa. (Jeronimo & Weast 2003, 151-153; UPnP Forum 2000, Wikipedia 2010.)

On huomioitava, että web-käyttöliittymän tekeminen UPnP-laitteille on täysin vapaaehtoista. Käyttöliittymän rakentamisessa on myös paljon vapauksia, sillä käyttöliittymien sisältöä ei ole standardisoitu millään tavalla UPnP foorumin puolesta. Käyttöliittymän rakentajalla on täysi vapaus, millainen käyttöliittymä on ulkoasultaan ja mitä ominaisuuksia käyttöliittymään liitetään. UPnP Forumin mukaan, jos jokin laite tarjoaa osoitteen käyttöliittymään laitekuvauksessaan, on sen myös annettava osoite sitä pyydettäessä. Käyttöliittymä pitää saada ladatuksi Internet-selaimeen onnistuneesti. Myös käyttöliittymän HTML-koodi tulee olla vähintään HTML versio 3.0 tai sitä uudempi. (Jeronimo & Weast 2003, 152-153.)

Presentation-protokollassa on myös huolehdittu lokalisaatioista, eli siitä, että käyttöliittymää voidaan käyttää kielestä riippumatta. UPnP-arkkitehtuuri nojaa HTTP:n valmiiseen lokalisaatiomekanismiin. Mekanismin avulla laite saa tietoonsa, mitä kieliä tuetaan ohjauspisteen tai web-selaimen päässä. Tällä tiedolla UPnP-laite voi lähettää web-sivun, joka käyttää tuettuja kieliä. Mekanismissa on käytössä kaksi HTTP-headeria, Accept-Language ja Content-Language. Kun selain tai ohjauspiste pyytää käyttöliittymäsivua, se käyttää Accept-Language-headeria. Accept-Language-headeri kertoo sen, mitä kieliä käyttöliittymää pyytävä taho tukee. UPnP-laitteen saatua HTTP-pyynnön, se lukee Accept-Language-headerin tiedon. Laitteen vastauksessa käytetään Content-Language-headeria, joka kertoo, mitä kieltä on käytetty web-sivussa, joka lähetetään. Jos laite tukee Accept-Language-headerissa pyydettyä kieltä, laite valitsee kyseisen kielen Content-Language-headeriin. (Jeronimo & Weast 2003, 156-159.)

Presentation-protokollassa on käytössä erilaiset merkistöjen enkoodaukset. Yleiset merkistöt UPnP:ssä ovat US-ASCII, ISO-8859-1, UTF-8 sekä UTF-16. Oletusmerkistönä on UTF-8. Tieto käytetystä enkoodauksesta kerrotaan HTTP-headereissa. Headerit lähetetään HTTP-requestin mukana, jossa pyydetään käyttöliittymäsivua. Käytössä on kaksi headeria, Content-Type ja Charset. Molemmilla headereilla siis kerrotaan, mikä enkoodaus dokumentissa on käytössä. Headereista käytetään ainoastaan jompaakumpaa eikä molempia. Käytännön toteutus on molemmissa lähes sama. (Jeronimo & Weast 2003, 157-158.)

2.5 UPnP AV

Universal Plug and Play Audio & Video (UPnP AV) on UPnP:n alue, jolle on tehty kaikkein eniten laitekehitystä. Nykyajan kodeissa on monenlaisia eri elektroniikkalaitteita, jotka osallistuvat äänen ja videon toistamiseen. Tällaisia ovat dvd-soitin, videonauhuri, tietokone, stereot yms. Jokainen laite toimii jollain tavalla äänen tai videon toistamisessa, mutta jokainen laite toimii pelkästään yksin. Laitteilla

ei myöskään ole tukea UPnP:n kaltaiseen teknologiaan, jossa etsitään, löydetään sekä ohjataan laitteita. UPnP AV:n tarkoitus on ratkaista nämä ongelmat. UPnP AV tarjoaa arkkitehtuurin äänen sekä videon jakamiseen käyttäen UPnP:tä. (Jeronimo, Weast 2003, 359-360.)

UPnP AV esittelee kaksi uutta komponenttia UPnP-teknologiaan. Komponentit ovat mediapalvelin sekä mediatoistin. Mediapalvelimen tarkoitus on toimia varastona kaikelle medialle. Mediana voi olla esimerkiksi musiikkia, videoita tai kuvia. Mediatoistimen tarkoitus on toistaa mediapalvelimissa sijaitsevia mediakohteita. Mediapalvelin ja mediatoistin voivat olla fyysisesti kokonaan eri laitteita. (Jeronimo, Weast 2003, 359-360.)

UPnP AV:n toiminta tapahtuu hyvin samalla tavalla, kuin muussa UPnP:ssä, jossa laitteita ohjaavat ohjauspisteet. UPnP AV:ssä puhutaan vuorovaikutuksen triangelista, jossa osapuolina on ohjauspiste, mediapalvelin sekä mediatoistin. Ohjauspiste etsii mediapalvelimen ja mediatoistimen. Ohjauspiste käyttää tiettyjä UPnP-toiminoja yhdistääkseen mediapalvelimen ja mediatoistimen siten, että laitteet voivat kommunikoida suoraan keskenään ilman, että ohjauspiste olisi kommunikaation välissä. Toimintojen avulla mediapalvelin voi esimerkki lähettää virran johonkin mediakohteeseen, jota mediatoistin alkaa toistaa. (Jeronimo, Weast 2003, 360; Steinfeld 2010.)

UPnP Forum on määrittänyt tietyt spesifikaatiot, jotka mediapalvelimien ja mediatoistimien on noudatettava. Spesifikaatioissa on mm. palvelut ja toiminnot, jotka palvelimen ja toistimien on pakko toteuttaa. Pakollisissa palveluissa ja toiminnoissa on määritelty vähimmäisvaatimukset laitteen toimimiselle. Mediapalvelimilla on yksi pakollinen palvelu, ContentDirectory. ContentDirectory-palvelu antaa toiminnot mediapalvelimen sisällön tutkimiselle. Mediatoistemalla on pakollinen palvelu nimeltä RenderingControl. RenderinControl-palvelu antaa toiminnot määrittää asetukset, miten mediaa toistetaan. Palvelussa on esimerkiksi toiminnot kirkkauden ja äänenvoimakkuuden säätämiseen. Lisäksi molemmilla komponenteilla on kaksi

yhteistä palvelua, pakollinen ConnectionManager ja valinnainen AVTransport. ConnectionManager-palvelu sisältää toiminnot mediapalvelimen ja mediatoistimen väliseen kommunikaatioon. Palvelussa on esimerkiksi toiminto, jolla saadaan selville laitteiden tukemat yhteydet kommunikaatiolle. AVTransport-palvelu tuo toiminnot mediapalvelimilla sijaitsevien mediakohteiden toistamiselle. Palvelu sisältää esimerkiksi toiminnot median toistamiselle ja pysäyttämiselle. AVTransport-palvelu on valinnainen, mutta käytännössä sen pois jättäminen halvauttaa laitteen. Esimerkiksi mediatoistin ei voi toistaa mediapalvelimilla olevia mediakohteita, jos AVTransport-palvelu ei ole käytössä. (Jeronimo, Weast 2003, 361-381; Steinfeld 2010.)

3 OHJELMOINTITYÖN KULKU

Tässä luvussa kerrotaan ohjauspisteen ohjelmoinnista. Luvun alussa kerrotaan ohjelmointityön taustoja ja sen jälkeen mennään itse ohjelmointiin.

3.1 Taustaa

Ohjelmointikieleksi valitsin Javan. Sovellusympäristönä käytettiin Java Micro Editionia (Java ME). Ohjelmointiympäristöksi valitsin Eclipsen, koska minulla oli siitä jo useamman vuoden kokemus. Ohjelmointityön alussa tutkin ohjelmointirajapintaa, joka oli rakennettu Java Standard Editionille (Java SE). Rajapinta oli Satoshi Konnon kehittämä CyberLink for Java. Yritin muuttaa rajapintaa niin, että se toimisi Java ME:llä. Huomasin kuitenkin muuntamisen olevan lähes mahdotonta. Rajapinta käytti niin paljon Java SE:n omia luokkia, jolloin nekin olisi pitänyt liittää mukaan mobiiliversioon. Samoin osa luokista oli sellaisia, että ne olivat käytössä molemmissa – Java SE:ssä ja Java ME:ssä. Ne kuitenkin erosivat toisistaan siinä, että Java ME-versio on hieman karsitumpi. Näiden luokkien tuominen mobiiliversioon edellyttäisi Java ME:n luokkien korvaamista Java SE-versioilla. Tämä osoittautui mahdottomaksi toteuttaa, sillä se rikkoisi koko Java ME:n luokkahierarkian.

UPnP arkkitehtuurin mukaan UPnP-laitteiden ja ohjauspisteiden tulee lähettää viestejä toisilleen käyttäen apuna multicastingia. Discovery-vaiheessa ohjauspisteet lähettävät discovery-viestin kaikille UPnP-verkon osapuolille ja kaikki verkossa olevat laitteet vastaavat ohjauspisteelle. Laitteet taas lähettävät viestejä jokaiselle verkon ohjauspisteille, kun laitteet tulevat verkkoon tai poistuvat verkosta. Nämä kaikki viestit lähetetään käyttäen IP-multicastia, eli viesti lähetetään jokaiselle verkon osapuolelle.

Törmäsin uuteen ongelmaan koskien ohjelmointiosuutta. Java ME ei tue verkkoliikenteessä IP multicastia, joka on suuressa roolissa UPnP:ssä. Ilman multicastia ohjauspiste ei voisi saada tietoon verkossa olevia muita laitteita. Multicastin tuominen Java SE:stä oli myös mahdotonta, koska multicast-luokat olivat riippuvaisia muista luokista. Tuli tarve tehdä systeemi, joka ei käytä multicastia, mutta kuitenkin on yhteydessä muihin laitteisiin.

UPnP-arkkitehtuurin mukaan jokaisen UPnP laitteen sekä ohjauspisteen välinen kommunikointi tapahtuu suoraan laitteiden välillä. Kommunikaatiossa ei siis käytetä mitään välikäsiä. Kommunikaatio tapahtuu HTTP:n avulla.

Kolme hollantilaista opiskelijaa, Bouke Pieter de Vries, Sijmon Heitmeijer sekä Jorrit Praamstra, olivat myös työskennelleet UPnP:n parissa. He olivat tehneet yhdessä ratkaisun, joka mahdollisti UPnP:n mobiilipuolella ilman, että tarvitsi IP-multicastia. He olivat ratkaisseet ongelman siten, että tietokoneella toimi ohjauspiste, joka toimi myös palvelimena mobiililaitteelle. Ohjauspiste keräsi tiedon kaikista verkossa olevista laitteista ja mobiililaite sai tiedon laitteista ohjauspisteen kautta. Ohjauspiste käytti samaa ohjelmointirajapintaa, jonka olin itsekin löytänyt. Ohjauspisteeseen oli lisäksi rakennettu yhteydet mobiililaitteen kanssa kommunikoimiseen. Mobiililaite yksinkertaisesti otti yhteyden ohjauspisteeseen ja sai sieltä listan laitteista. Toimintojen suorittaminen mobiililaitteella toimi siten, että mobiililaitteelta lähetettiin viesti ohjauspisteelle ja ohjauspiste tulkitsi viestistä, mikä toiminto suoritetaan ja sen jälkeen toiminto suoritettiin. Mobiililaite ei siis itse suorittanut toimintoja, vaan ohjauspiste suoritti toiminnotkin. Minulla oli ratkaisun lähdekoodit saatavilla, joten aloitin sen läpikäymisen. Selasin ja tutkin koko lähdekoodin kauttaaltaan. Ymmärsin, että voin käyttää tätä samaa ratkaisua omassa työssänikin pienin muutoksin.

Kuitenkin tämä ratkaisu, jossa mobiililaite kommunikoi tavallaan välikäden kautta, ei toteuta UPnP arkkitehtuurin määrittelyjä. Mobiililaite ei keskustele suoraan minkään laitteen kanssa, vaan kaikki tapahtuu palvelimena toimivan ohjauspisteen kautta.

Kuitenkin tämä on ainoa järkevä ratkaisu Java ME:n vajavaisista ominaisuuksista johtuen.

Ohjelmointityössä tarvittiin myös mediapalvelinta sekä mediatoistinta oman ohjelman testaamiseen. Tein päätöksen, että molemmat, sekä palvelin, että toistin, pyörii samalla tietokoneella, millä tein ohjelmointiakin. Valitsin mediapalvelimeksi Satoshi Konnon tekemän Cyber Media Gaten. Mediatoistimena käytin Foobar2000:ta, johon oli asennettu UPnP-plugin.

3.2 Ohjelmoinnin toteutus

Tässä luvussa kerrotaan ohjelmointityöstä. Luku on jaettu alalukuihin, joissa kussakin luvussa kerrotaan ohjelmointityön eri vaiheista. Luvussa vuorottelee käytännön työ sekä teoria.

3.2.1 Informointi jokaisesta laitteesta

Ohjelmointini toteutuksen lähtökohtana oli hollantilaisten tekemä tekniikkademo, joka kuvasi, miten UPnP:llä voidaan ohjata valokatkaisinta. Demoon oli tehty kaksi omaa UPnP-laitetta, jotka edustivat valoja, ohjauspiste, joka toimi samalla mobiililaitteen palvelimena sekä itse mobiililaite, jolla ohjattiin valolaitteita. Tekniikkademon ohjauspiste oli rakennettu niin, että se kertoi mobiililaitteelle ainoastaan demon omista valolaitteista. Ensimmäiseksi muutin ohjauspistettä niin, että se informoi mobiililaitetta jokaisesta löydetystä laitteesta. Tämä kävi helposti muokkaamalla vain yhtä ohjauspisteen metodia, joka valitsi ne laitteet, joista informoitiin mobiililaitteelle. Käytännössä metodista tarvitsi poistaa vain muutama tarkastus ja sen jälkeen mobiililaite sai tiedon kaikista laitteista.

3.2.2 Käyttöliittymän muokkaus

Mobiililaitteen käyttöliittymä oli rakennettu pelkästään valolaitteiden ohjaamiseen, joten käyttöliittymää oli muokattava. Käyttöliittymässä oli yhteensä neljä eri ikkunaa: Ensimmäisessä ikkunassa voi valita ohjauspisteen, jonka kanssa halutaan keskustella. Toisessa ikkunassa tulee kirjoittaa käyttäjätunnus ja salasana. Kolmas ikkuna näyttää löydetyt laitteet. Viimeinen ikkuna on varsinainen laitteen ohjausikkuna, jota tuli muokata. Poistin kaikki valmiina ikkunassa olevat painikkeet ja korvasin ne uusilla. Painikkeita tuli tässä työvaiheessa kolme: pause, stop ja play.

UPnP AV arkkitehtuurin mukaan mediatoistimilla on valinnainen palvelu, jonka avulla mediatoistin voi toistaa mediakohteita, jotka sijaitsevat mediapalvelimilla. Palvelu on nimeltään AVTransport. Mediatoistimien palvelujen tulisi tarjota toimintoja, joilla voidaan ohjata mediatoistimen tilaa siten, että onko mediatoistin toisto-tilassa vai pause-tilassa.

Painikkeista puuttui vielä toiminnallisuus kokonaan, joten se oli työn seuraava vaihe. Ensimmäiseksi tuli selvittää, mitä UPnP-toimintoja tulee laukaista laitteessa, jotta saadaan pause, -stop –ja play-painikkeet toimimaan. Haluamani toiminnot olivat juuri samannimiset, mitä painikkeetkin, Pause, Stop ja Play (UPnP Forum 2008). Toiminnot löytyivät AVTransport-palvelusta, joka on valinnainen palvelu mediatoistimille. Palvelu löytyi myös Foobar2000:sta. Sitten piti saada tietoon, mitä argumentteja kukin toiminto tarvitsee. Play-toiminnolla on kaksi argumenttia: InstanceID ja Speed. Pause ja stop-toiminnoilla on yksi ja sama argumentti, InstanceID (UPnP Forum 2008).

3.2.3 Mobiililaitteen ja ohjauspisteen välinen kommunikaatio

UPnP-laitteiden ohjaaminen tulisi teorian mukaan tapahtua siten, että ohjauspisteet lähettävät SOAP-viestejä laitteille. SOAP-viestien data on XML-dokumentissa. XML-

dokumentista saadaan selville mm. mitä toimintoa halutaan laukaista, mitä argumentteja halutaan muuttaa ja mitkä ovat muutettavien argumenttien uudet arvot. Ohjauspisteet lähettävät viestit suoraan halutulle laitteelle HTTP:n yli.

Mobiililaitteen uusien painikkeiden toiminnallisuuden toteutus vaati muutoksia sekä mobiilipuolella, että ohjauspisteessä. Alkuperäinen mobiilipuolen ratkaisu toimi UPnP-toimintoja laukaisevien painikkeiden osalta niin, että painiketta painaessa kutsutiin metodia, jolla oli merkkijono parametrina. Metodi lähetti ohjauspisteelle parametrina tulleen merkkijonon. Saatuaan viestin ohjauspiste muunsi merkkijonon UPnP-toiminnoksi. Tämän jälkeen ohjauspiste teki toiminnosta SOAP-viestin ja lähetti sen laitteelle. UPnP-toiminto oli ohjelmointirajapinnan Action-luokan instanssi. Luokalla voitiin määritellä toimintoon liittyvät argumentit sekä kyseisten argumenttien arvot. Luokalla oli myös metodi, jolla luotiin toiminnosta SOAP-viesti ja viesti lähetettiin laitteelle.

Tämä toteutus ei ole UPnP:n standardien mukainen. Jos haluttaisiin, että mobiililaite toimisi oikeasti ohjauspisteenä, tulisi sen lähettää SOAP-viestit suoraan laitteille. Tässäkin toteutuksessa käytetään palvelimena toimivaa ohjauspistettä välikätenä. Kuitenkin tällainen toteutus on toimiva ja osaltaan sallittu, koska mobiililaite tarvitsee joka tapauksessa palvelimena toimivaa ohjauspistettä.

Mobiilipuolella tuli määritellä ohjauspisteelle lähetettävä merkkijono uudelleen. Päädyin ratkaisuun, jossa tein jokaisesta toiminnosta oman luokan. Luokalla on metodi, joka kutsuu alkuperäistä metodia, joka siis lähettää viestin ohjauspisteelle. Luokka sisältää myös merkkijonomuuttujan, jota käytetään alkuperäisen metodin parametrina. Ohjauspisteen puolella tuli tehdä muutoksia sen metodin osalta, joka muunsi mobiililaitteelta saadun viestin UPnP-toiminnoksi eli Action-luokan instanssiksi. Ratkaisin metodin kirjoittamalla kokonaan uuden metodin, joka teki juuri samat asiat, mutta kuitenkin haluamallani tavalla. Kirjoittamani metodi käytti edelleen samaa ohjelmointirajapintaa.

3.2.4 Sisällön haku mediapalvelimelta

UPnP AV-arkkitehtuurin mukaan mediapalvelimet varastoivat mediaa, kuten kuvia, ääntä sekä videoita. Arkkitehtuurin mukaan mediapalvelimien sisältöä pitäisi päästä jotenkin tutkimaan. Jokaisen mediapalvelimen tulisi toteuttaa palvelut, joiden avulla ohjauspisteet pääsevät tutkimaan mediapalvelimien sisältöä.

Mobiililaitteella ei voinut vielä selata mediapalvelimien sisältöä. Selauksen toteuttamiseksi tuli selvittää UPnP-toiminnot, joilla voidaan pyytää palvelimelta sisältöä. Haluttu toiminto oli nimeltään Browse (UPnP Forum 2008). Browse-toiminnolla voi pyytää mediapalvelimelta sisältöä ja palvelin lähettää vastauksen sisällön kansiorakenteesta. Vastaus on XML-dokumenttina. Huomasin, että tarvitsen kahta erilaista versiota Browse-toimminnosta: Toiminto, joka hakee kansiorakenteen juurikansion ja Toiminto, joka hakee halutun kansion alikansiot sekä muut mediakohteet. Browse-toiminnolla oli suuri määrä argumentteja. Sisään menevät argumentit olivat ObjectID, BrowseFlag, Filter, StartingIndex, RequestedCount sekä SortCriteria (UPnP Forum 2008). Ulospäin menevät argumentit olivat Result, NumberReturned, TotalMatches, sekä UpdateID (UPnP Forum 2008). Kuitenkin minun tuli ottaa huomioon vain argumentit ObjectID, BrowseFlag sekä Result. Jokaisella mediapalvelimen kansiolla on tunnistusavain, jonka avulla eri kansiot erotetaan toisistaan. Tämä avain on toiminnoissa argumentti ObjectID. Alikansioiden sisältöjen haussa tarvittiin ObjectID-argumenttia. Argumentin arvoksi laitettiin sen kansion tunnistusavain, jonka alikansiota haluttiin hakea. BrowseFlag-argumentilla määriteltiin käytännössä, haettiinko palvelimen juurikansiota, vai alikansioita. Argumenttiin kirjoitettiin BrowseMetadata, kun haluttiin hakea juurikansiota ja kun haettiin alikansioita, argumenttiin kirjoitettiin BrowseDirectChildren. Result-argumentti oli ulospäin menevä argumentti, joka oli käytännössä vastaus haluttuun hakuun. Result-argumentti oli XML-dokumentti, joka sisälsi haun tuloksen. Lisäsin ohjauspisteen metodiin, joka laukaisee play, pause ja stop toiminnot, myös mahdollisuuden tehdä hakutoimintoja mediapalvelimilta.

Sisällön hakutoiminto tapahtui ohjauspisteessä, jolloin haun vastauksenkin sai ohjauspiste. Vastaus XML-dokumenttina tuli saada XML-muodosta sellaiseen muotoon, jonka mobiililaitekin ymmärtää. XML:n muokkaamiseen käytin apuna Ajay Vohran ja Deepak Vohran kirjoittamaa kirjaa Pro XML Develoment with Java™ Technology. Kirjassa esiteltiin esimerkki XML-dokumentin jäsennöimisestä, jonka avulla sain tehtyä oman metodin, joka jäsennöi mediapalvelimelta tulleen vastauksen. XML:n muokkaamisen käytin Java SE:n valmiita Document Object Model-luokkia, joiden avulla vastaus saatiin XML-dokumentista merkkijonoksi. Merkkijonoa muokattiin vielä siihen suuntaan, että merkkijonosta voidaan puhelimen päässä erotella vastauksen eri kohteet toisistaan. Merkkijono oli muotoa tyyppi#nimi@parametri|. Merkkijonon #, @ sekä |-merkit olivat apumerkkejä, joiden avulla mobiililaitteessa merkkijono voidaan tulkata niin, että siitä erotetaan vastauksen kohteet toisistaan. Merkkijonossa tyyppi tarkoittaa kohteen tyyppiä, eli onko vastaus kansio vai mediakohde. Nimi on kansion tai mediakohteen nimi. Parametri riippuu vastauksen tyypistä. Jos tyyppi on kansio, on parametri silloin kansion tunnistusavain, edellä mainittu ObjectID-argumentti. Jos tyyppi on mediakohde, parametrina on osoite suoraan kyseiseen mediakohteeseen. Osoitetta käytetään hyväksi silloin, kun halutaan soittaa haettua mediakohdetta jossain mediasoittimessa.

Ohjauspisteen ja mobililaitteen välinen kommunikointi tapahtui siten, että ohjauspiste välitti tietoa löydetyistä laitteista ja mobiililaite lähetti pyyntöjä laitteiden toimintojen laukaisemiseksi. Nyt tuli tarve saada mobiililaitteelle tieto hakutoimintojen vastauksista. Muutin mobiilaitteessa metodia, joka lukee ohjauspisteeltä tulleen viestin. Lisäsin metodiin tarkistuksen, onko viesti päivitys löydetyistä laitteista vai onko viesti vastaus hakutoiminnosta. Viestistä riippuen, ohjelmassa kutsutaan metodia, joka vie saadun viestin jatkokäsittelyyn.

Hakutoiminnon tuloksen saaminen näkyviin mobiililaitteessa vaati vielä muutoksia mobiililaitteen ohjelmaan. Ensimmäisenä lisäsin uuden painikkeet samaan ikkunaan, jossa pause, play ja stop painikkeet jo olivat. Painikkeet nimeksi tuli Browse media.

Seuraavaksi tein painikkeen painallukselle toiminnallisuuden. Tein uuden ikkunan, joka avautuu painiketta painaessa. Ikkunan tarkoitus on tulostaa näytöllä hakutulokset, jotka on tehty halutulle mediapalvelimelle. Laitteen käyttöliittymä oli tehty siten, että tässä vaiheessa käyttäjä on jo joutunut valitsemaan halutun mediapalvelimen, josta sisältöä haetaan. Tämän tietäessäni rakensin ikkunan avautumisen siten, että samaan aikaan kun ikkuna avautuu, puhelin on jo lähettänyt ohjauspisteelle viestin tehdä juurikansion hakutoiminto. Ikkunan avautuessa ikkunaan tulostuu siis ensimmäisen haun tulos, mediapalvelimen juurikansio. Tulostuksen tein siten, että ruutuun lisätään pelkästään joukko painikkeita, jossa yksi painike edustaa aina yhtä kansiota tai mediakohdetta. Painikkeet luodaan ikkunan update-metodissa, joka tulkkaa ohjauspisteeltä saadun viestin. Metodi lukee viestistä, montako kohdetta vastauksessa on ja rakentaa silmukassa yhtä monta painiketta. Silmukassa luodaan myös painikkeille toiminnallisuus. Painikkeen toiminta riippuu siitä, edustaako painike kansiota vai mediakohdetta. Painikkeen edustaessa kansiota, lähetetään ohjauspisteelle pyyntö suorittaa hakutoiminto, jossa haetaan kyseisen kansion sisältöä, eli alikansioita tai mediakohteita. Jos painike edustaa mediakohdetta, otetaan mediakohteesta ylös osoite, jota tarvitaan haluttaessa soittaa ko. mediakohdetta jossain mediatoistimessa.

3.2.5 Mediapalvelimen ja –toistimen välinen kommunikaatio

UPnP AV-arkkitehtuurin mukaan mediapalvelimella ja mediatoistimella tulee olla palvelut, joiden avulla palvelin ja toistin voivat kommunikoida toistensa kanssa. Palvelun tarkoitus on luoda yhteys näiden kahden pisteen välille. Yhteyden avulla mediapalvelin voi esimerkiksi lähettää virtana musiikkikappaleen, jonka mediatoistin toistaa. Arkkitehtuurin mukaan mediapalvelimen ja mediatoistimen tulee toteuttaa ConnectionManager-palvelu. Palvelu sisältää toiminnot mediapalvelimen ja mediatoistimen välisen yhteyden saamiseksi.

Tässä vaiheessa mobiililaitteella voitiin käskeä mediasoitinta soittamaan musiikkia ja pysäyttämään soitto hetkellisesti tai kokonaan. Mobiililaitteella voitiin myös tutkia mediapalvelimien sisältöä jokaista kansiota myöten. Kuitenkin yksi toiminto vielä puuttui. Mobiililaitteen soittamisen aloitus ja pysäytys koski pelkästään sitä kappaletta, joka oli jo valittuna mediatoistimessa. Tuli tarve tehdä toiminnallisuus soittaa mediatoistimissa juuri se kappale, joka oli valittu mediapalvelimien sisällön etsimisen yhteydessä. Mediatoistimet tarvitsevat mediakohteen osoitteen, jota halutaan soittaa. Ilman osoitetta mediatoistimet eivät voi soittaa musiikkia. Tarvittiin UPnP-toiminto, joka kertoo mediatoistimelle mediakohteen linkin. Tällainen toiminto on nimeltään SetTransPortURI (UPnP Forum 2008). Toiminnolla on kolme argumenttia, InstanceID, CurrentURI, CurrentURIMetaData. CurrentURI on argumentti, johon kirjoitetaan mediakohteen osoite (UPnP Forum 2008). Kun tämä toiminto suoritetaan mediatoistimella, toistin vaihtaa soitettavaa kappaletta. Tämän toiminnon jälkeen voidaan suorittaan esimerkiksi Play-toiminto, jonka jälkeen haluttua mediakohdetta aletaan toistaa. Kyseisen toiminnon tuomiseksi ohjelmaan mobiililaitteen puolella luotiin jälleen uusi luokka kyseiselle toiminnolle. Mobiililaitteen ikkunaan lisättiin vielä yksi painike, SetTransportUri, jolla lähetetään ohjauspisteelle viesti, että halutaan suorittaan ko. toiminto. Ohjauspisteessä muutettiin viimeistä kertaa metodia, joka tulkkaa mobiililaitteen lähettämästä viestistä sen, mitä toimintoa halutaan laukaista laitteessa. Metodiin siis lisättiin vielä vaihtoehto SetTransPortURI-toiminnolle.

3.2.6 Testaus

Ohjelmoinnin viimeisenä työvaiheena oli ohjelmien testaus ja koodin dokumentointi. Ohjelmien toimintaa testattiin jokaisen työvaiheen jälkeen. Lopussa tehtiin ohjelmien laajempi testaus, sillä tähän asti testaus oli tehty vain emulaattorilla. Ohjelmaa testattiin myös varsinaisessa puhelimessa. Puhelimeen asennettiin ohjelmisto ja testattiin ohjelmiston toimivuutta. Testauksen jälkeen tein katsauksen kirjoittamaani

koodiin. Kommentoin joitakin kommentoimattomia metodeja sekä poistin koodista turhia koodirivejä. Ohjelmisto oli valmis.

4 TULOKSET

Tässä luvussa kerrotaan työn tuloksista. Luvun ensimmäisessä alaluvussa selostetaan, mitä ohjelmalla voi tehdä ohjelmoinnin ansiosta. Kahdessa muussa kappaleessa selostetaan PC:llä toimivan ohjauspisteen sekä mobiilisovelluksen toiminnan kuvaukset.

4.1 Ohjelmointityön tulokset

Ohjelmointityön saavutus on, että mobiililaitteella voi ohjata mediapalvelimia ja mediatoistimia. Ohjaus ei kata kaikkia ominaisuuksia, mutta kriittisimmät toiminnot on tuettuna. Mobiililaitteella voi selata mediapalvelimien sisältöä juurikansiosta aina viimeisempään kansioon asti. Mobiililaite listaa kaikki löydetyt kohteet, oli ne sitten mediakohteita tai kansioita. Selaamisen lisäksi mobiililaiteella voi valita jonkin mediakohteen, joka löytyy mediapalvelimelta. Valitun mediakohteen voi sitten toistaa halutussa mediatoistimessa. Näiden lisäksi mobiililaitteesta löytyy painikkeet, jolla ohjataan mediatoistimien toistotilaa, eli toistaako mediatoistin kappaletta, vai onko toistin pause- tai stop-tiloissa.

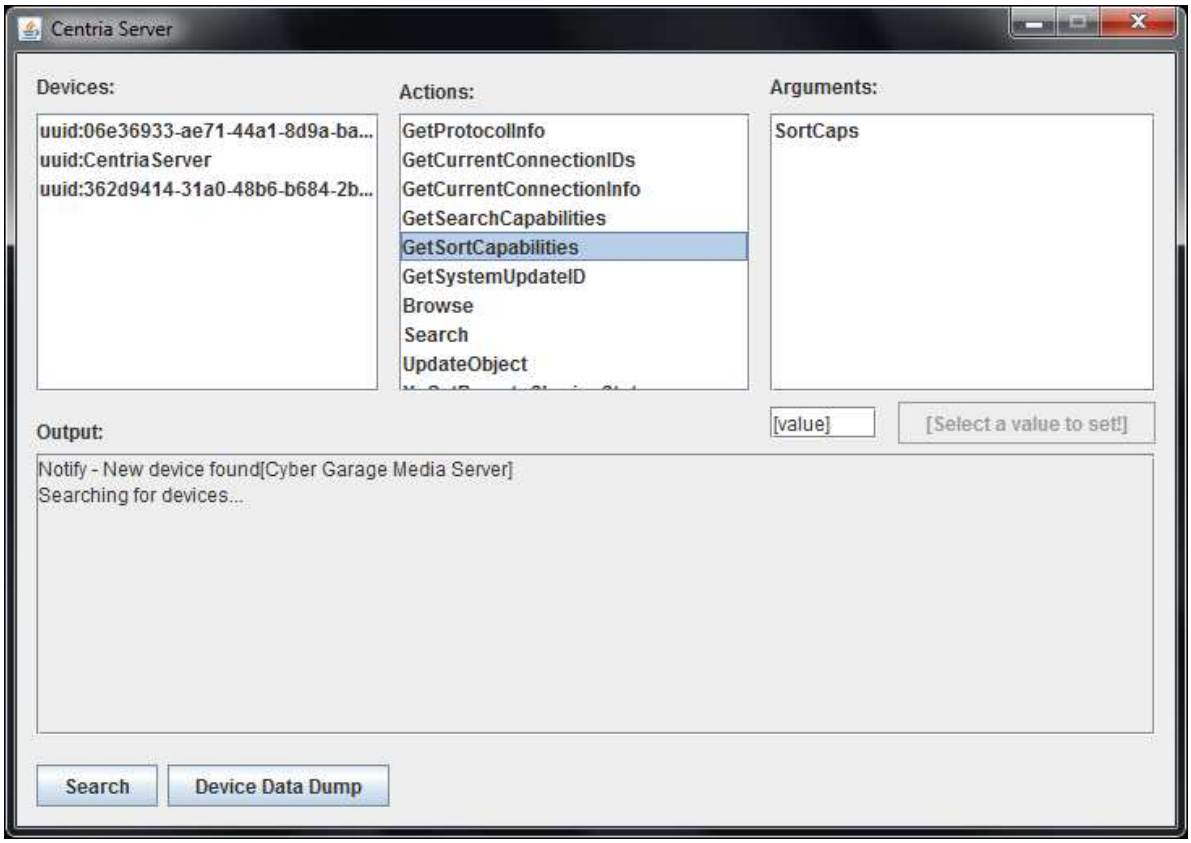
Mobiililaitteesta jäi kuitenkin uupumaan joitakin ominaisuuksia. UPnP mahdollistaa toiminnot next- ja previous-toiminnoille. Näillä toiminnoilla voi siis valita soitettavaksi seuraavan tai edellisen kappaleen. Lisäksi mobiililaitteen käyttöliittymää voisi parantaa huomattavasti. Esimerkiksi kappaleen valitseminen käyttöliittymästä ja soittaminen mediatoistimessa on suhteellisen monen napin painalluksen päässä. Lisäksi koko ohjelmistoa testattiin pelkästään musiikkikappaleiden osalta. Testauksen voisi myös kuvilla ja äänillä.

4.2 PC:llä toimiva ohjauspiste

Ohjauspisteessä on käytössä pelkästään yksi ikkuna. Ikkunassa 4 ruutua: Output, Devices, Actions sekä Arguments. Output listaa ohjelman kaikki tapahtumat. Tapahtumia on mm. uusien laitteiden löytö verkosta, informoiminen mobiililaitteelle laitteista yms. Aina kun jokin tapahtuma tapahtuu, output-ruutuun tulee tekstinä tietoa tapahtumasta. Devices-ruutu listaa kaikki verkosta löydetyt laitteet. Ruutuun tulostetaan löydettyjen laitteiden UUID. Actions-ruutu listaa valitun laitteen toiminnot. Toiminnot tulostetaan vain sillon, kun on hiirellä painettu jotain laitetta Devices-ruudussa. Arguments-ruutu tulostaa valitun toiminnon argumentit. Argumentit tulostetaan silloin, kun jotain toimintoa klikataan Actions-ruudusta.

Ohjelmassa on vielä kolme painiketta sekä yksi tekstilaatikko. Painikkeet ovat Search, Device Data Dump sekä Select value to set. Search-painikkeella tehdään haku, jolla haetaan verkossa olevia UPnP-laitteita. Device Data Dump-painike tulostaa kaikkien löydettyjen laitteiden tiedot Output-ruutuun. Select value to set-painike toimii vain silloin, kun Arguments-ruudussa jokin argumentti on valittuna. Painikkeella voidaan asettaa arvo halutulle argumentille. Ohjelman tekstilaatikkoon kirjoitetaan haluttu arvo ja sen jälkeen voidaan painaa Select value to set-painiketta.

Ohjelman muu toiminnallisuus, kuten laitteiden löytäminen, listaaminen Devices ruutuun ja tietojen lähettäminen mobiililaitteelle, tapahtuu automaattisesti. Kuviossa 1 on kuvakaappaus ohjauspisteestä.



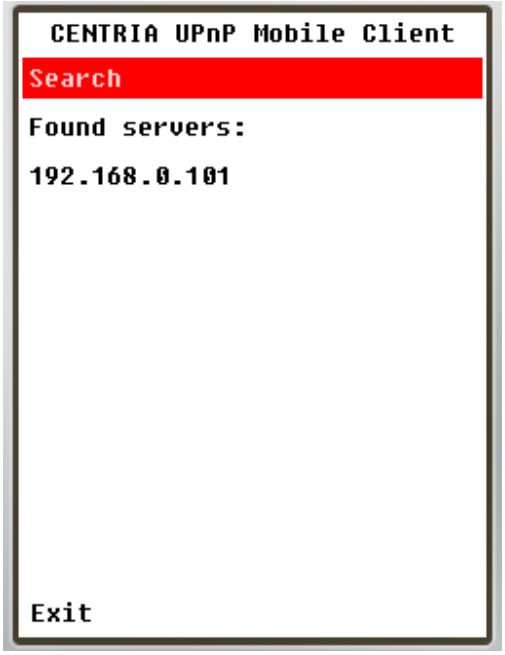
KUVIO 1. Kuvakaappaus PC:llä toimivasta ohjauspisteestä.

4.3 Mobiilisovellus

Mobiilisovellus koostuu lähinnä ikkunoista ja painikkeista. Jokainen ikkuna täyttää koko puhelimen ruudun. Ohjelma kuvataan kronologisesti alkaen ohjelman ensimmäisestä ikkunasta

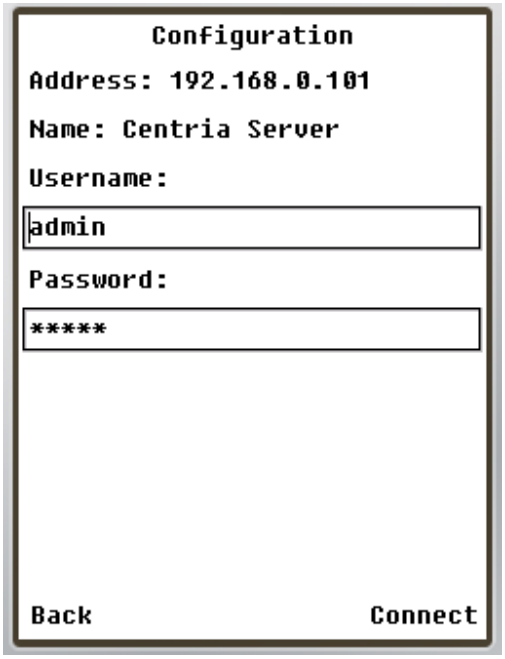
Ensimmäisessä ikkunassa valitaan löydetty ohjauspiste, jonka kanssa halutaan keskustella. Ohjauspiste on siis käytännössä edellisessä luvussa esitelty ohjauspiste, joka toimii myös palvelimena mobiililaitteelle. Ikkunassa on painike search, jolla voidaan tehdä haku, jolla haetaan nimenomaan tätä palvelimena toimivaa ohjauspistettä. Löydetyt ohjauspisteet listataan search-painikkeen alle. Jokaiselle löydetylle pisteelle luodaan ikkunassa oma painike ja painikkeen nimenä on löydetyn

ohjauspisteen IP-osoite. Ikkunassa on myös Exit-painike, jolla poistutaan koko ohjelmasta. Kuvio 2 on ensimmäisestä ikkunasta. Kuvassa on löytynyt yksi ohjauspiste. Ohjauspisteen painiketta painaessa siirrytään toiseen ikkunaan.



KUVIO 2. Mobiilisovelluksen ensimmäinen ikkuna.

Toisessa ikkunassa kysytään käyttäjätunnusta ja salasanaa. Ikkunassa on tekstinä ohjauspisteen nimi sekä ohjauspisteen IP-osoite. Näiden tekstien alla on kaksi tekstilaatikkoa, joihin tulee kirjoittaa käyttäjätunnus ja salasana. Koko ikkuna demonstroi UPnP-turvallisuutta. Demonstraatiolla osoitetaan, että laitteiden ohjaamiseksi tulee tietää käyttäjätunnus ja salasana. Muuten laitteita pääsisi ohjaamaan kuka tahansa. Kuitenkin ikkuna on vain demonstraatio, sillä käyttäjätunnukseen ja salasanaan voi kirjoittaa mitä tahansa ja ohjelma hyväksyy sen. Ikkunan alla on painikkeet Back ja Connect. Back-painikkeella päästään takaisin edelliseen ikkunaan ja Connect-painikkeella otetaan varsinaisesti yhteys ohjauspisteeseen. Connect-painiketta painattaessa siirrytään ohjelman kolmanteen ikkunaan. Kuvio 3 on kuvankaappaus toisesta ikkunasta.



KUVIO 3. Mobiilisovelluksen toinen ikkuna.

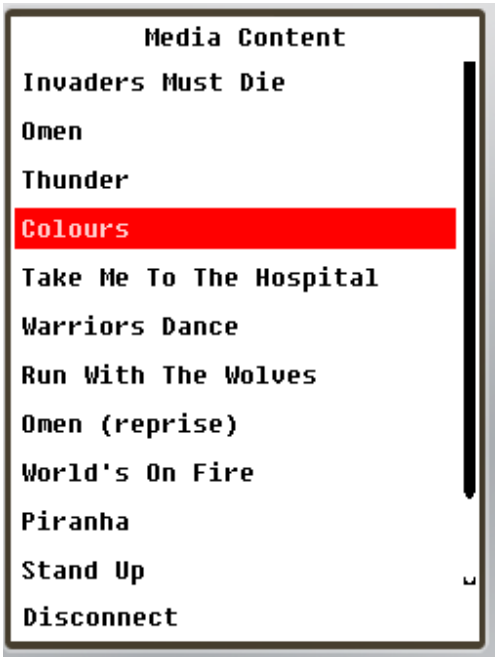
Mobiilisovelluksen kolmas ikkuna listaa kaikki UPnP-laitteet, jotka ohjauspiste on löytänyt ja informoinut mobiililaitteelle. Ikkuna koostuu kokonaan painikkeista, joista yksi painike edustaa aina yhtä löydettyä laitetta. Painikkeiden nimet muodostuvat sitä edustavan laitteen nimestä. Kun jotain näistä painikkeista painetaan, siirrytään ohjelman neljänteen ikkunaan, jossa voidaan jo hallita UPnP-laitteita. Ikkunassa on myös Disconnect-painike, jota painamalla siirrytään mobiilisovelluksen ensimmäiseen ikkunaan. Kuvio 4 on kuvankaappaus sovelluksen kolmannesta ikkunasta.

KUVIO 4. Mobiilisovelluksen kolmas ikkuna.

Neljännessä mobiilisovelluksen ikkunassa voidaan ohjata UPnP-laitteita. Ikkunan nimi on edellisessä ikkunassa valitun laitteen nimi. Ikkunassa on viisi painiketta: Play, Pause, Stop, SetAVTransportURI sekä Browse media. Play, Pause sekä Stop-painikkeilla voidaan laukaista Play, Pause ja Stop toiminnot valitussa UPnP-laitteessa. SetAVTransportURI-painikkeella voidaan laukaista SetAVTransportURI-toiminto valitussa laitteessa. Browse media-painikkeella laukaistaan Browse-toiminto. Browse-painiketta painattaessa siirrytään ohjelman viimeiseen eli viidenteen ruutuun. Ikkunassa on myös Back-painike, jolla siirrytään ohjelman edelliseen ikkunaan, jossa voidaan valita laite. Kuvio 5 esittää ohjelman neljättä ikkunaa.

KUVIO 5. Mobiilisovelluksen neljäs ikkuna.

Mobiilisovelluksen viidennessä ikkunassa voidaan hakea mediapalvelimien sisältöä. Käytännössä ruudussa on pelkästään painikkeita. Kukin painike edustaa joko kansiota tai mediakohdetta. Kun painike edustaa kansiota, painiketta painattaessa ikkuna pyyhkiytyy ja näkyville tulee valitun kansion sisällä olevaa sisältöä. Jos taas painike edustaa mediakohdetta, painiketta painattaessa siirrytään takaisin sovelluksen neljänteen eli edelliseen ruutuun. Ikkunassa on myös Back-painike, jolla siirrytään sovelluksen kolmanteen ruutuun, jossa laitteet on listattuna. Kuvio 6 on kuvankaappaus viidennestä ikkunasta. Ikkunassa tutkitaan erästä musiikkialbumin sisältöä. Sisältönä on koko albumin kappaleet.



KUVIO 6. Mobiilisovelluksen viides ikkuna.

On huomioitava, että jos haluaa soittaa jossain mediatoistimessa musiikkikappaleen, jonka on valinnut mobiilisovelluksella, tulee toimia seuraavasti. Ensiksi on valittava mediapalvelin. Mediapalvelimella on haettava sisältöä. Halutun kappaleen löydyttyä on kappaleen painiketta painettava. Sen jälkeen on siirryttävä takaisin laitevalikkoon ja valittava laite, joka toimii mediatoistimena. Laitteen valinnan jälkeen on painettava SetTransportURI-painiketta, ja sen jälkeen on painettava Play-painiketta. Tällä tavoin saadaan haluttu musiikkikappale soimaan.

5 POHDINTA

Lopullinen ohjelmisto ei vastaa aivan täysin UPnP:n suunniteltua arkkitehtuuria. Arkkitehtuurin mukaan ohjauspisteet kommunikoivat laitteiden kanssa suoraan eikä minkään välikäden kautta. Tämä ei toteudu ohjelmistossa, sillä mobiililaite kommunikoi muiden laitteiden kanssa ohjauspisteen kautta, joka toimii palvelimena mobiililaitteelle. Palvelimena toimiva ohjauspiste suorittaa kaikki UPnP-toiminnot laitteille ja pitää listaa verkossa olevista laitteista. Ohjauspiste informoi mobiililaitteelle kaikista verkon laitteista ja kuuntelee jatkuvasti mobiililaitteelta tulevia viestejä. Mobiililaite osallistuu UPnP-laitteiden ohjaamisen siten, että mobiililaite lähettää viestin palvelimena toimivalle ohjauspisteelle ja ohjauspiste tulkkaa lähetetystä viestistä, mikä toiminto halutaan laukaista. Tällainen ratkaisu ei ole UPnP arkkitehtuurin määritysten mukainen. Kuitenkin Java ME:n vajavaisten ominaisuuksia vuoksi oli mahdotonta rakentaa mobiililaitteelle ohjelmistoa, joka kommunikoisi suoraan muiden laitteiden kanssa. Palvelimena toimiva ohjauspiste noudattaa täysin UPnP:n arkkitehtuurin määrityksiä. Se toimii itsenäisenä ohjauspisteenä, joka kommunikoi suoraan muiden laitteiden kanssa. Ohjauspisteeseen on vain lisätty ominaisuus mobiililaitteen kanssa kommunikoimiseen.

Työn tavoitteina oli rakentaa toimiva ohjauspiste sekä saada kokonaisvaltainen käsitys UPnP:n arkkitehtuurista. Mielestäni työn tavoitteet täyttyivät suurimmalta osalta. Vaikka mobiililaitteen ratkaisu ei täysin vastaakaan UPnP:n arkkitehtuuria, on kuitenkin parempi, että se toimii hieman sääntöjä rikkoen kuin, että se ei toimisi ollenkaan. Muitakin tavoitteita jatkokehitykseen vielä jäi, mm. jokainen kappale piti erikseen valita. Käytännön toteutuksesta jäi siis next- ja previous–toiminnot pois kokonaan. Kuitenkin mobiililaitteella toimii kaikkein tärkeimmät toiminnot eli mediapalvelimien sisällön selaaminen sekä haluttujen kappaleiden soittaminen halutulla mediatoistimella. Tavoitteet täyttyivät myös arkkitehtuurin ymmärtämisen osalta myös. Arkkitehtuuri tuli tutuksi kokonaisuudessaan ennen ohjelmointityön alkua ja ymmärrys kasvoi ohjelmointityön aikana.

Ohjelmistoon jäi vielä paljon parannettavaa jatkokehityksen varalle. Mobiililaitteen käyttöliittymä on varsin kankea käytettävä. Käyttöliittymää voisi parannella esimerkiksi siten, että mediakohteen valinnassa voitaisiin välittömästi valita myös, missä mediatoistimessa valittu kappale soitetaan. Nykyisessä ohjelmistossa kappaleen valinnan jälkeen on mentävä takaisin laitevalintaikkunaan, valita mediatoistin ja sen jälkeen painaa play-painiketta. Lisäksi ohjelmaan voitaisiin lisätä jo edellisissä kappaleissa mainitut next- ja previous-toiminnot. Koko ohjelmisto on kokonaisuudessaan hyvä pohja minkä tahansa UPnP-laitteen kehittämiseen ja sen ohjaamiseen. Mobiililaitteella tarvitsee tehdä muutoksia vain laitteen ohjausikkunassa, jossa lähetetään viestejä ohjauspisteelle. Ohjauspisteellä tarvitaan muutoksia sen metodin osalta, joka lukee mobiililaitteelta tulleen viestin.

Työ oli kokonaisuudessaan erittäin mielenkiintoinen. En ollut ennen työn aloittamista kuullutkaan koko UPnP:stä. Työssä riitti myös paljon haasteita, koska ohjelmointia tuli tehdä mobiililaitteelle, josta minulla oli varsin vähän kokemusta. Kokemukseni karttui huomattavasti ohjelmoinnin osalta. Mobiililaitteen ja ohjauspisteen välisen kommunikoinnin toteutus toi etenkin minulle paljon kokemusta, sillä en ollut ennen opiskellut tietoliikenteen toteuttamisia ohjelmoinnissa. Samoin XML oli minulle uusi asia, johon en ollut perehtynyt ennen opinnäytetyötä. Työssä oli myös haastetta sopivasti. Ohjelmointi ei ollut liian helppoa, eikä liian vaikeaakaan. Moniin ongelmiin törmäsin ohjelmoinnin aikana, mutta jokaiseen ongelmaan löysin ratkaisu. Työ oli kokonaisuudessaan erittäin mukava, haastava ja ennen kaikkea mielenkiintoinen. Voi olla, ettei tämä ole viimeinen kerta, kun työskentelen UPnP:n parissa.

LÄHTEET

Jeronimo, M, Weast, J. 2003. UPnP Design by Example.
Yhdysvallat: Richard Bowles

Vohra, A, Vohra, D. 2006. Pro XML Development with Java Technology, 41-48. E-Kirja. Saatavissa: http://books.google.fi/books?id=pQ2h64OBSSAC&printsec=frontcover&dq=Pro+xml+development+with+java&cd=1#v=onepage&q&f=false. Luettu 20.4.2010.

Wikipedia: UPnP. 2010. Www-dokumentti. Saatavissa: http://en.wikipedia.org/wiki/UPnP. Muutettu: 29.3.2010. Luettu: 20.4.2010.

Wikipedia: Zero configuration networking. 2010. Www-dokumentti. Saatavissa: http://en.wikipedia.org/wiki/Zero_configuration_networking. Muutettu: 16.4.2010. Luettu: 20.4.2010.

Wikipedia: Ad-hoc network. 2010. Www-dokumentti. Saatavissa: http://en.wikipedia.org/wiki/Ad-hoc_network. Muutettu: 11.4.2010. Luettu: 20.4.2010.

Wikipedia: LinkLocal. 2010. Www-dokumentti. Saatavissa: http://en.wikipedia.org/wiki/LinkLocal. Muutettu: 4.4.2010. Luettu: 20.4.2010.

Konno, S. 2005. Cyber Link for java programming guide. Pdf-tiedosto. Saatavissa: http://sourceforge.net/projects/cgupnpjava/files/clinkjava-doc/1.7/clinkjavaproguide170.pdf/download. Muutettu: 29.5.2005. Luettu: 19.3.2010.

Steinfeld. E. Home Entertainment Automation Using UPnP AV Acrhitecture and Technology. Pdf-tiedosto. Saatavissa: http://www.go-embedded.com/UPnP%20White%20Paper.pdf. Luettu: 20.4.2010.

UPnP Forum. 2008. AVTransport:2 Service Template Version 1.01. Pdf-tiedosto. Saatavissa: http://www.upnp.org/specs/av/upnp-av-avtransport-v2-service-20080930.pdf. Muutettu: 30.9.2008. Luettu: 30.3.2010.

UPnP Forum. 2000. Understanding UPnP™: A White Paper. Doc-tiedosto. Saatavissa: http://www.upnp.org/download/UPNP_UnderstandingUPNP.doc. Muutettu: 6.2000. Luettu: 11.4.2010.

UPnP Forum. 2008. UPnP™ Device Architecture 1.1. Pdf-tiedosto. Saatavissa: http://www.upnp.org/specs/arch/UPnP-arch-DeviceArchitecture-v1.1.pdf. Muutettu: 15.10.2008. Luettu: 10.4.2010

UPnP Forum. 2008. ContentDirectory:3 Service Template Version 1.01. Pdf-tiedosto. Saatavissa: http://www.upnp.org/specs/av/upnp-av-contentdirectory-v3-service-20080930.pdf. Muutettu 30.9.2008. Luettu 31.3.2010