



Arto Karppinen

Irreco: kauko-ohjausta Internet Tabletilla

Arto Karppinen

Irreco: kauko-ohjausta Internet Tabletilla

Arto Karppinen

Opinnäytetyö

Kevät 2011

Tietojenkäsittelyn koulutusohjelma

Oulun seudun ammattikorkeakoulu

TIIVISTELMÄ

Oulun seudun ammattikorkeakoulu
Tietojenkäsittelyn koulutusohjelma

Tekijä: Arto Karppinen
Opinnäytetyön nimi: Irreco: kauko-ohjausta Internet Tabletilla
Työn ohjaaja: Jouni Juntunen
Työn valmistumislukukausi ja -vuosi: Kevät 2011 Sivumäärä: 41

Opinnäytetyön lähtökohtana on kehittää Nokia 770 ja Nokia N800 -laitteissa toimiva kauko-ohjausohjelmisto, jolla käyttäjä voi ohjata viihde-elektroniikkalaitteita. Ohjelmistoon rakennetaan kaksi erilaista tilaa joissa ohjelmisto toimii. Editointitilassa käyttäjä voi rakentaa itselleen käyttöliittymän viihde-elektroniikkalaitteiden ohjaamista varten. Käyttötilassa ohjelmisto näyttää editointitilassa rakennetun käyttöliittymän, jonka avulla käyttäjä voi ohjata viihde-elektroniikkalaitteita.

Kohdeympäristönä on Nokian Maemo-käyttöjärjestelmä, joka pohjautuu Debian Linux -käyttöjärjestelmään. Ohjelmointikielenä on C. Ohjelmoinnissa on käytetty GTK+- ja GLib-kirjastoja. IR-signaalien lähettämiseen ja käsittelyyn on käytetty LIRC-nimistä ohjelmistoa. Ohjelmistoa kehitetään ns. avoimen lähdekoodin projektina. Ohjelmisto julkaistaan GPL-lisenssin alaisuudessa ja ohjelmiston lähdekoodit julkaistaan Maemo Garage -sivustolla.

Lopputuloksena syntynyt ohjelmisto, Irreco, täyttää ohjelmalle asetetut vaatimukset.

Asiasanat: Linux, Maemo, GTK+, ohjelmointi, kauko-ohjaus, avoin lähdekoodi

ABSTRACT

Oulu University of Applied Sciences
Business Information Technology

Author: Arto Karppinen
Title of thesis: Irreco: remote control with an Internet Tablet
Supervisor: Jouni Juntunen
Term and year when the thesis was submitted: Spring 2011 Pages: 41

The purpose of this Bachelor's thesis is to develop a remote control application for Nokia 770 and Nokia N800 devices, which user can use to control various consumer electronic devices. Two distinct modes will be built into the application. In edit mode the user of the application can build an user interface for controlling consumer electronic devices. In use mode the application displays the user interface created in edit mode, which user can then use to control his or hers consumer electronic devices.

Target environment for the application is Nokia's Maemo operating system, which is based on Debian Linux operating system. Programming is done in C language. GTK+ and GLib libraries are used in the application. For sending and receiving IR signals a software called LICR is used. The application is developed as an open source project. The application will be released under GPL license and source code of the application will be released on a website called Maemo Garage.

The application produced as a result, Irreco, fullfills the requirements set for the application.

Keywords: Linux, Maemo, GTK+, programming, remote control, open source

SISÄLLYS

1 Johdanto	6
1.1 Lähtökohdat	9
1.2 Suunnitelmat	9
2 Ympäristö	11
2.1 Avoimen lähdekoodin historiaa lyhyesti	11
2.2 Linux-distribuutiot	13
2.3 Maemo	14
2.4 Scrachbox	16
2.5 GTK+- ja GLib-kirjastot	17
2.6 LIRC	19
3 Irreco	21
3.1 Käyttöliittymäeditori	21
3.2 Näppäimet	23
3.3 Fyysiset painikkeet	26
3.4 Komentoketjut	27
3.5 Sisäiset komennot	28
3.6 Teemat	29
3.7 Käyttötila	31
4 Komentojärjestelmät	33
4.1 Backend-kirjastojen toteutuksesta	34
4.2 LIRC-backend	35
5 Pohdinta	38
Lähteet	41

1 JOHDANTO

Nokia julkaisi toukokuussa 2005 Nokia 770 -laitteen. Nokia 770 on Linux-käyttöjärjestelmään pohjautuva langaton internet-päätelaite, eli Internet Tablet, jota on ensisijaisesti markkinoitu käyttäjille helppona tapana käyttää internet palveluita missä ja milloin tahansa. Laitteen mukana tulevat sovellukset sisältävätkin esimerkiksi internet-selaimen, pikaviestimen ja sähköpostin. (Kuva 1.)



KUVA 1. Nokia 770

Kun laitetta tarkastellaan tekniseltä tasolta, on Nokia 770 käytännössä pieni Linux-tietokone, joka voi tehdä periaatteessa kaikkea sitä mitä tavallinenkin Linux-tietokone. Tämän takia laitteen ohjelmistotarjonta onkin laajentunut suuresti pelkän internetkäytön ulkopuolelle. Kaikkein Nokia 770 ei kuitenkaan kykene, sillä pieni näyttö ja hitaahko prosessori asettavat rajoituksia.

Tässä opinnäytetyössä kehitetään Nokia 770 -laitteessa toimiva ohjelmisto, nimeltään Irreco, joka mahdollistaa erilaisten viihde-elektroniikkalaitteiden kauko-ohjauksen. Ongelmana nykyajan viihde-elektroniikkalaitteiden ohjauksessa on usein niiden määrä, sillä

moneen kotitalouteen on hankittu ainakin televisio, digiboxi ja DVD-soitin. Koska jokaisen laitteen mukana tulee oma kaukosäädin, on laitteiden ohjaaminen tarpeettoman hankalaa. Irrecon avulla käyttäjä voi hallita kaikkia näitä laitteita Nokia 770 -laitteesta käsin, sen sijaan että hänen tarvitsisi käsitellä jokaisen laitteen kaukosäädintä erikseen.

Viihde-elektroniiikan kaukosäätimet viestivät yleensä lähettämällä sähkömagneettista säteilyä, jota kutsutaan infrapunaksi tai infrapunavaloksi. Yleensä kuitenkin käytetään lyhennettä IR, joka tulee englanninkielen sanasta infrared. Painettaessa kaukosäätimen painiketta lähettää kaukosäädin sarjan valopulsseja, jotka kohdelaitteessa oleva valoherkkä diodi vastaanottaa ja muuttaa sähkövirraksi. Kauko-ohjaus on myös mahdollista toteuttaa muillakin tavoin, esimerkiksi radioaaltoja hyväksikäyttäen, mutta näitä mahdollisuuksia ei huomioida tässä opinnäytetyössä.

Koska Nokia 770 ei sisällä IR-lähetintä, ei sillä voi suoraan lähettää IR-signaaleja viihde-elektroniikkalaitteille. Laitteesta kuitenkin löytyy tuki WLAN:lle, eli langattomalle lähiverkolle, jonka avulla se voi olla yhteydessä internetiin ja toisiin tietokoneisiin. Langattoman lähiverkon kautta Nokia 770 -laitteesta voidaan ottaa yhteys Linux-tietokoneeseen, johon on asennettu LIRC-niminen ohjelmisto, joka osaa lähettää ja vastaanottaa IR-signaaleja. Koska LIRC-ohjelmistoa on mahdollista etähallita lähiverkon kautta, voi Irreco ohjata erilaisia viihde-elektroniikkalaitteita käyttämällä LIRC-ohjelmistoa välikätenä. (Kuva 2.)



KUVA 2. Irrecon toimintaperiaate

1.1 Lähtökohdat

Tässä opinnäytetyössä keskitytään nimenomaisesti rakentamaan ohjelmiston käyttöliittymää ja käyttöliittymäeditoria, jolla käyttäjä voi rakentaa Nokia 770 -laitteen ruudulle haluamansa käyttöliittymän viihde-elektroniikan ohjaamiseen. Ohjelmointikielenä on C ja ohjelmointiympäristönä Nokian Maemo-käyttöjärjestelmä. Käytännössä tämä tarkoittaa C-ohjelmointia GTK+- ja GLib-kirjastoja käyttäen. IR-signaalien lähettämiseen käytetään LIRC-nimistä ohjelmistoa, jolloin Irrecon ei tarvitse huolehtia IR-signaalien matalantason käsittelystä. Tästä syystä tässä opinnäytetyössä ei myöskään käsitellä infrapunajärjestelmän matalantason toteutusta.

Opinnäytetyön toisena lähtökohtana on luoda Irrecoasta avoimen lähdekoodin projekti ja pyrkiä muodostamaan yhteisöä sen ympärille. Tämä tarkoittaa sitä, että ohjelmiston lähdekoodi ja muut materiaalit julkaistaan Maemo Garagessa, joka on Nokian ylläpitämä avoimen lähdekoodin projektisivusto Internet Tablet -ohjelmistokehittäjille.

1.2 Suunnitelmat

Työn toimeksiantajana oli Nokia. Projekti yksi useista opiskelijaprojekteista, joihin Nokia otti opinnoissaan pitkällä olevia opiskelijoita kehittämään hyödyllisiä apuohjelmia Nokia 770 -laitteelle. Tässä opinnäytetyössä kuvattu osuus projektista oli aktiivinen vuosina 2007 ja 2008. Tämän jälkeen Irreco on laajennettu muiden henkilöiden toimesta.

Ohjelmiston alkuperäisenä vaatimuksena oli rakentaa ohjelmisto, jonka avulla käyttäjä voi lisätä kosketusnäytölle painikkeita haluamaansa järjestykseen. Käyttäjä voisi sen jälkeen yhdistää painikkeet haluamiinsa komentoihin ja painikkeen painaminen käskyttäisi LIRC-ohjelmistoa lähettämään oikean IR-signaalin jollekin laitteelle.

Viihde-elektronikkalaitteiden ohjaus oli tarkoitus hoitaa LIRC-ohjelmiston avulla, niin ettei Irrecon tarvitsisi huolehtia IR-signaalien syvällisemmästä käsittelystä. Toisin sanoen, Irreco olisi vain käyttöliittymä, jonka avulla käyttäjä pääsisi helposti käsiksi LIRC-ohjelmiston toimintoihin.

Myöhemmin toimeksiantajan vaatimukset laajenivat koskemaan laitteen fyysisten painikkeiden tukemista, erilaisia painiketyylejä, tukea useille kaukosäätimille ja erilaisten kauko-ohjausjärjestelmien tukemista.

Ohjelmiston kehityksen aikana myös sain käyttööni Nokia N800 -laitteen, joka on Nokia 770 -laitteen päivitetty versio. Koska N800 ei merkittävästi eroa Nokia 770 -laitteesta, päätin vaihtaa N800-laitteen ohjelmiston kehityskohteeksi.

2 YMPÄRISTÖ

2.1 Avoimen lähdekoodin historiaa lyhyesti

Avoimen lähdekoodin historia juontaa juurensa vuosiin 1971 – 1985, jolloin Richard Stallman niminen ohjelmoija oli töissä MIT:ssä, eli Massachusetts Institute of Technologyssä. MIT:n AI-laboratoriossa Stallmanin työnä oli lisätä PDP-10 tietokoneen ITS-käyttöjärjestelmään ominaisuuksia. AI-laboratorio oli hieno oppimisympäristö, jossa ohjelmistojen kehitettiin ystävyyden, tieteellisen avoimuuden, hakkerimaisen nokkeluuden ja kilvoittelun ilmapiirissä. Stallman kuvaakin aikaa erääksi hänen elämänsä onnellisimmista. 1980 luvun alkupuolella suurin osa AI-laboratorion työntekijöistä palkattiin yksityisyrittäjien työntekijöiksi ja Stallmanin suuresti rakastama hakkeriyhteisö hajosi. MIT myös antoi yritykselle oikeuden tehdä liiketoimintaa ja jatkokehittää LISP-nimistä ohjelmaa, joka oli kehitetty AI-laboratoriossa. Samaan aikaan ITS-käyttöjärjestelmähanke oli kaatumassa, sillä AI-laboratorioon oli tulossa uusi tietokone, eikä AI-laboratoriossa enää ollut pätevää henkilökuntaa siirtämään sitä uudelle tietokoneelle. (Moody 2001, 28 - 31)

Nämä tapahtumat järkyttivät Stallmania syvästi ja hän päätti aloittaa oman käyttöjärjestelmänsä rakentamisen, joka olisi kaikille avoin ja vapaa, eikä sitä voitaisi sulkea yrityksen omaisuudeksi. Näistä lähtökohdista alkoi kehittyä GNU, eli "GNU's Not Unix!". GNU-käyttöjärjestelmän nimi on rekursiivinen akronyymi, eli akronyymi mikä sisältää itsensä. Kyseessä on tyypillinen esimerkki hakkerihuumorista jota harrastettiin AI-laboratoriossa. (Moody 2001, 32 - 35)

Kuitenkin kenties tärkein asia, joka syntyi näiden tapahtumien tuloksena oli GPL, eli Gnu Public License. Useimmat ohjelmistolisenssit on suunniteltu suojelemaan yrityksen oikeuksia ja varmistamaan yrityksen tulonsaanti ohjelmistojen myymällä. Tämä johtaa lisensseihin

jotka rajoittavat käyttäjän oikeuksia. Erityisesti ohjelmiston lähdekoodi on tarkasti vartioitu yrityssalaisuus, jota ei missään nimessä julkaista tai näytetä ulkopuolisille. Tämä tekee ohjelmiston muokkauksen mahdottomaksi ja estää käyttäjiä tekemästä omia versioita ohjelmasta. Koska Stallmanin tavoitteet olivat huomattavan erilaiset kuin perinteisillä yrityksillä, hän tarvitsi oman lisenssin. Ajan mittaan lopputulokseksi muotoutui GPL, joka on suunniteltu toimimaan päinvastaisella tavalla kuin perinteiset lisenssit. GPL antaa käyttäjille kaikki oikeudet muokata lähdekoodia, mutta ei salli lähdekoodin sulkemista. Toisin sanoen GPL on suunniteltu varmistamaan ohjelmiston avoimuus, käyttäjien vapaus muokata sitä, sekä estää mahdolliset kolmansien osapuolien haltuunottoyritykset. GPL-lisenssillä on ollut suuri vaikutus avoimen lähdekoodin mahdolliseksi tekemisessä, ja tuhansia ohjelmistoja onkin julkaistu sen alaisuudessa. Esimerkiksi Sourceforge.net listaa 110 090 projektia jotka käyttävät GPL-lisenssiä (Search, Sourceforge.net, Hakupäivä 18.3.2011).

Loppuen lopuksi vuosien työn tuloksena Stallmanilla oli vuonna 1990 kasassa kaikki käyttöjärjestelmän tärkeimmät ohjelmat, paitsi yksi, kerneli. Kerneli on käyttöjärjestelmän ydin, joka kontrolloi tietokoneen laitteistoa. Tämän viimeisen puuttuvan palan tarjosi lopulta nuori suomalainen ohjelmoija nimeltä Linus Torvalds. Torvalds oli käyttänyt Minix-nimistä opetuskäyttöön suunniteltua Unixin tapaista käyttöjärjestelmää. Torvalds kuitenkin halusi ohjelmoida juuri saamalleen 386-tietokoneelle oman käyttöjärjestelmän oppiakseen tietokoneen toiminnasta, ja koska koki sen mielenkiintoiseksi haasteeksi. Niinpä Torvalds ryhtyi kehittämään Linuxia käyttäen Minixiä työympäristönä. Ajan mittaan Torvalds ohjelmoi oman kernelinsä kaikki osat kokoon ja julkaisi Linuxin ensimmäisen version vuonna 1991 lyhyen itse kirjoitetun lisenssin alaisuudessa. Pari versiota myöhemmin vuonna 1992 Torvalds julkaisi käyttäjien pyynnöstä Linuxin GPL-lisenssin alaisuudessa. (Moody 2001, 55, 61 - 67, 73.)

Yhdessä Linux-kerneli ja GNU-projektin ohjelmistot muodostavat kokonaisen käyttöjärjestelmän. Tästä syystä jotkin Linux-distribuutiot kutsuvat itseään nimellä GNU/Linux, mikä havainnollistaa molempien projektien tärkeyttä. Yleensä kuitenkin puhutaan pelkästään Linux-käyttöjärjestelmistä tai Linux-distribuutioista.

2.2 Linux-distribuutiot

Koska Linux ja GNU ovat avoimen lähdekoodin projekteja, on mahdollista tehdä monenlaisia valintoja siitä, mitä komponentteja Linuxiin pohjautuvaan käyttöjärjestelmään pitäisi kuulua, ja miten niiden tulisi toimia. Linux-yhteisössä tämä on johtanut kehitykseen, jossa erilaisia Linuxiin ja GNU:hun pohjautuvia käyttöjärjestelmiä on syntynyt useita kappaleita. Näitä erilaisia versioita Linux-käyttöjärjestelmästä kutsutaan nimellä distribuutio, suomeksi jakelu, yleensä puhekielessä distro.

Jakelut eroavat toisistaan käyttökohteiden, teknisten ratkaisujen ja ideologisten mielipiteiden perusteella. Erilaisia distribuutioita on satoja, aina palomuuridistribuutiosta, kotiteatteridistribuutioon ja palvelindistribuutioon. Esimerkiksi DistroWatch.comin haku listaa satoja erilaisia Linux-distribuutioita. Tarkkaa määrää on kuitenkin vaikea saada, sillä sivusto näyttää myös muihin avoimen lähdekoodin kerneleihin, kuten BSD:hen, perustuvia projekteja. Tämän lisäksi sivustolla on myös mukana kuolleita distribuutioita. Tästä huolimatta, erilaisia hakuja käyttämällä on mahdollista haarukoida aktiivisten Linux-distribuutioiden lukumäärä noin kolmeensataan.

Distribuutioiden laajuus myös vaihtelee suuresti. Pienimmät mahtuvat yhdelle levykkeelle, suurimmat sisältävät kymmenien tuhansien ohjelmistojen kirjastoja, joista käyttäjät voivat ladata haluamansa ohjelmat Internet-yhteyden läpi. Esimerkiksi Debianin kirjastosta löytyy 35906 eri pakettia. (All Debian Packages in "squeeze", Debian.org. Hakupäivä 31.3.2011.)

Suurimmat ja eniten Linuxin kehitykseen vaikuttaneet distribuutiot lienevät Debian ja Red Hat. Suurin osa muista distribuutioista perustuu joko Debianiin tai Red Hat:iin lisäen, poistaen tai muokaten niitä jollakin tavalla jotain tiettyä päämäärää silmälläkatsoen.

Debian on vapaan ja avoimen lähdekoodin ympärille muodostunut yhteisö, jonka ideologiana on kehittää kokonaan vapaata ja kaikille avointa käyttöjärjestelmää. Debian toimii käytännössä kokonaan vapaaehtoisvoimin ja lahjoitusten turvin. (About Debian, Debian.org. Hakupäivä 31.3.2011.)

Red Hat on samannimisen yrityksen kehittämä distribuutio, jota kehitetään erityisesti palvelinkäyttöä ja yritysten tarpeita varten. Red Hat rahoittaa toimintansa yritysten ostamien tuki- ja ylläpitopalveluiden maksuista. (Services & Products, RedHat.com. Hakupäivä 31.3.2011.)

Viimevuosina suurta suosiota työpöytäkäytössä saavuttanut Ubuntu on Debianiin pohjautuva distribuutio, joka lisää Debianin pohjalle työpöytäkäyttöä helpottavia muutoksia ja kolmansien osapuolien kirjoittamia suljettuja ohjelmistoja, kuten esimerkiksi AMD:n ja NVIDIAN näytönohjainajurit. Debianiin suljettuja ohjelmistoja ei hyväksytä ideologisista syistä. Ubuntu kehittää Canonical-niminen yhtiö, joka pyrkii rahoittamaan toimintansa tuki- ja ylläpitopalveluiden maksuista. (Ubuntu for your business, Ubuntu.com. Hakupäivä 31.3.2011.)

2.3 Maemo

Vaikka Maemoa käytetäänkin terminä hieman epäselvästi, on Maemo käytännössä Nokian Internet Tablet -laitteisiin suunnittelema Debian pohjainen Linux-distribuutio (Maemo Diablo Reference Manual for maemo 4.1, Maemo.org, 73. Hakupäivä 31.3.2011). Maemo-termiä käytetään myöskin kuvaamaan Internet Tablet -laitteiden käyttöjärjestelmää ja kehitysympäristöä, jolla kehitetään ohjelmistoja

näihin laitteisiin. Tässä opinnäytetyössä käytetään sanaa Maemo tarkoittamaan näiden kahden asian muodostamaa kokonaisuutta.

Nokia on myöskin käyttänyt Maemosta nimeä Internet Tablet Operating System, minkä perään on vielä lisätty vuosinumero. Esimerkiksi vuonna 2008 julkaistusta Maemosta on käytetty nimiä, ITOS2008, OS2008 ja Internet Tablet Operating System 2008. Tämä on valitettavan sekavaa.

Maemo-kehitysympäristö vaatii toimiakseen tietokoneen, johon on asennettu jokin työpöytäkäyttöön tarkoitettu Linux-distribuutio. Käytännössä suositeltavaa kuitenkin olisi käyttää joko Ubuntu tai Debiania. Syynä tähän on se, että Maemo, kuten myöskin Ubuntu, pohjautuvat Debianiin.

Tämä helpottaa ohjelmistokehitystä, sillä käyttämällä Debiania, tai Debian pohjaista Ubuntu, voidaan kehitysympäristö pitää mahdollisimman yhtenäisenä. Myöskin kaikki Maemo-ohjelmistokehitykseen käytettävät ohjeet ja ohjelmat pääsääntöisesti olettavat, että käytössä on joko Ubuntu- tai muu Debian-pohjainen Linux-distribuutio.

Varsinaisessa ohjelmointityössä Maemo-ympäristössä käytetään Linux-ympäristöissä yleistä GNU-työkaluketjua. GNU-työkaluketju koostu GCC-kääntäjästä, GNU Binutils -linkittäjästä, GDB-debugerista sekä aputyökaluista kuten GNU make, Autoconf, Autoheader, Automake ja Libtool. Näillä työkaluilla on mahdollista kehittää, kääntää ja debugata ohjelmistoja. (Maemo Diablo Reference Manual for maemo 4.1, Maemo.org, 42, 56. Hakupäivä 31.3.2011.)

2.4 Scrachbox

Ongelmana ohjelmistokehityksessä Maemo-ympäristölle on se, että Internet Tablet -laitteet käyttävät ARM-prosessoriarkkitehtuuria toisin kuin PC-tietokoneet, jotka käyttävät X86-prosessoriarkkitehtuuria. Tästä syystä Linux-tietokoneella ei voi suoraan kehittää ohjelmistoja Internet Tablet -laitteille. Ratkaisuksi tähän ongelmaan on kehitetty Scratchbox. (Maemo Diablo Reference Manual for maemo 4.1, 74. Hakupäivä 31.3.2011.)

Scratchbox on ristiinkääntöympäristö, joka mahdollistaa ohjelmistokehityksen ARM-prosessoreille X86-tietokoneen sisällä. Tämä on mahdollista käyttämällä Qemua. Qemu on avointa lähdekoodia oleva prosessoriemulaattori, joka pystyy emuloimaan ARM-prosessorin tukemia komentoja, eli käskykanta. (Maemo Diablo Reference Manual for maemo 4.1, Maemo.org, 17-18. Hakupäivä 31.3.2011.)

Pelkkä käskykannan emulointi ei kuitenkaan riitä ohjelmistokehitykseen. Lisäksi tarvitaan käännoympäristö, kirjastot, perusohjelmat sekä kaikki muut Linux-järjestelmän peruskomponentit, jotta ohjelmistokehitys olisi mahdollista. Eikä ympäröivän Linux-tietokoneen ohjelmistoja voi käyttää, koska ne voisivat olla väärää versiota, tai väärälle käskykannalle käännettyjä.

Tämän takia Scrachboxissa on target järjestelmä. Targetin, eli kohdejärjestelmän tai kohdeympäristön, sisälle asennetaan kokonainen Linux-käyttöjärjestelmä kaikkine ohjelmistoineen ja osineen. Targetin sisälle kirjaudutaan Scrachboxin login-komennolla, jonka jälkeen targetissa voi suorittaa ohjelmia siten kuin ne toimisivat esimerkiksi N800-laitteessa. (Maemo Diablo Reference Manual for maemo 4.1, Maemo.org, 19-20. Hakupäivä 31.3.2011.)

Targetteja voi luoda Maemo-ympäristön eri versiolle, ja jokaiselle versiolle on olemassa ARM- ja X86-targetit. X86-targetti voi kuulostaa

kummalliselta, onhan tarkoitus kehittää ohjelmistoja ARM-proessoreita käyttäviin Internet Tablet -laitteisiin. Etuna kuitenkin X86-targetin käyttämisessä on se, ettei emulointia tarvita. Tämä taas huomattavasti nopeuttaa ohjelmien kääntämistä ja suorittamista, jolloin ohjelmien testaaminen ja kehittäminen nopeutuu.

X86-targetin käyttäminen Maemo-ohjelmistokehitykseen on mahdollista, koska sen sisältämä Maemo-käyttöjärjestelmä sisältää samojen ohjelmien samat versiot kuin ARM-targetti. Ainoa ero on siinä, että nämä ohjelmistot on käännetty X86-arkkitehtuurille ARM-arkkitehtuurin sijaan. Tämän vuoksi X86-targetissa kehitetyn ohjelmiston toiminnan tulisi olla samanlainen myös ARM-targetissa. Periaatteessa on kuitenkin mahdollista, että toiminta ei ole identtistä arkkitehtuurien erilaisuuksien takia, tai koska jotain ARM-arkkitehtuurin ominaisuutta ei ole onnistuttu emuloimaan täydellisesti. Tässä opinnäytetyössä ei kuitenkaan todettu tämänkaltaisia ongelmia.

Käytännön ohjelmointityössä ohjelmistoa yleensä kehitetään X86-targetin sisällä, kunnes ohjelmisto viimein halutaan asentaa esimerkiksi N800-laitteeseen. Tällöin aktiiviseksi targetiksi vaihdetaan ARM-versio ja ohjelmisto käännetään uudelleen sekä luodaan asennustiedostot N800-laitetta varten. Näin hidasta emulointia tarvitsee tehdä mahdollisimman vähän.

2.5 GTK+- ja GLib-kirjastot

Kirjasto on kokoelma ohjelmistoissa käytettäviä funktiota. Kirjaston tarkoitus on antaa kirjaston sisältämä ohjelmakoodi muiden ohjelmien ja kirjastojen käyttöön. Tällä tavoin saman asian toteuttavaa koodia ei tarvitse kirjoittaa uudelleen jokaista ohjelmaa varten, vaan uudelleenkäytettävä ohjelmakoodi voidaan sijoittaa kirjastoon, jota vasten eri ohjelmat voidaan linkittää.

GTK+ on käyttöliittymäkirjasto, joka sisältää erilaisia luokkia ja

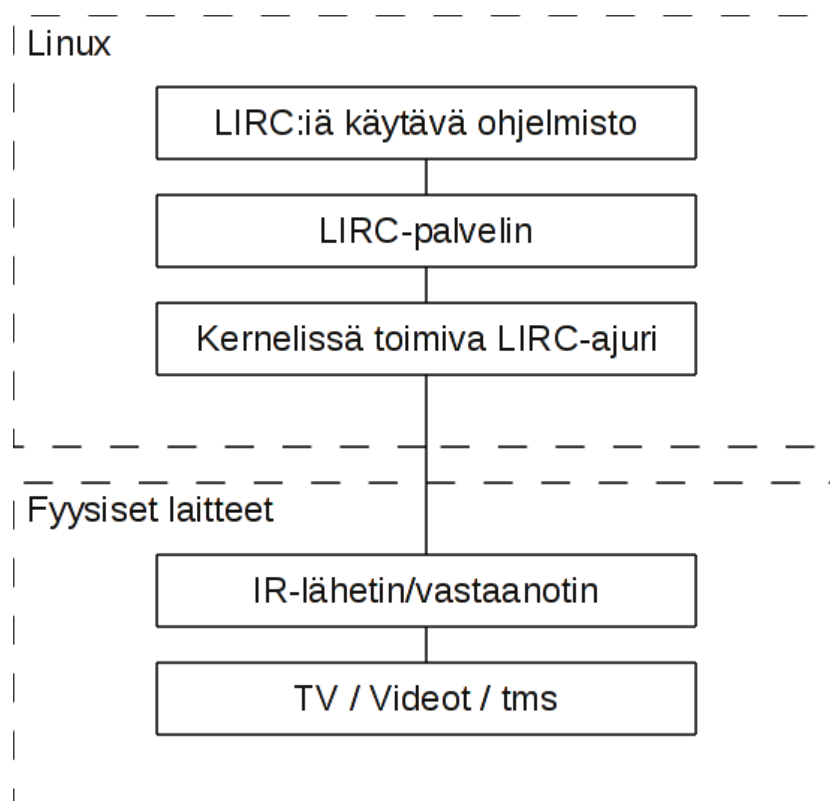
komponentteja ohjelmiston käyttöliittymän rakentamiseen. GTK+-kirjastoa käytetään yhdessä GLib-kirjaston kanssa. GTK+:n sisältäessä käyttöliittymäkomponentit, sisältää GLib erilaisia ohjelmoinnissa tarvittavia matalan tason funktiota ja toiminnallisuutta. Yhdessä GTK+ ja GLib muodostavat kokonaisuuden, jonka avulla ohjelmoija voi rakentaa kokonaisen sovelluksen ilman, että hänen tarvitsisi käyttää muita kirjastoja. Tämä on kätevää, sillä näin ohjelmointiympäristö pysyy yhteneväisenä ja helppokäyttöisenä. (GTK+ Overview, Gtk.org. Hakupäivä 25.4.2011.)

GTK+, eli GIMP Toolkit, kehitettiin alun perin GIMP nimisen kuvankäsittelyohjelman käyttöliittymän ohjelmointia varten vuonna 1998. Ajan mittaan GTK+ on kasvanut itsenäiseksi projektiksi ja on nykyään eräs keskeisimpiä avoimen lähdekoodin projekteja. GTK+- ja GLib-kirjastoja käytetään tuhansissa ohjelmistoissa ja koko Gnome työpöytäympäristö on rakennettu näitä teknologioita käyttäen. (GTK+ Overview, Gtk.org. Hakupäivä 25.4.2011.)

Luetun ymmärtämisen kannalta on hyvä mainita että, kirjaston nimi todellakin on GTK+. Siis plusmerkin kanssa. Kun tähän yhdistetään suomenkielen yhdyssanasäännöt väliviivan käytöstä, saadaan lopputulokseksi kummalliselta vaikuttavia kirjoitusasuja, kuten esimerkiksi GTK+-kirjasto.

2.6 LIRC

LIRC, eli Linux Infrared remote control, on ohjelmisto Linux-tietokoneisiin, joka mahdollistaa IR-signaalien vastaanottamisen ja lähettämisen. LIRC koostuu Linux-kerneliin ladattavista IR-vastaanottimien ja -lähettimien ajureista, palvelinohjelmistosta, joka välittää IR-komentoja ohjelmistoilta kernelille ja kerneliltä ohjelmistoille sekä asiakasohjelmistosta, jolla voidaan käskyttää palvelinta (Kuva 3).



KUVA 3. LIRC ohjelmiston toimintaperiaate

Koska LIRC osaa lähettää ja vastaanottaa IR-singnaaleja, voi tietokone, johon LIRC on asennettu, joko kauko-ohjata laitteita tai olla kauko-ohjauksen kohde. Tässä opinnäytetyössä LIRC:iä käytetään kuitenkin vain laitteiden kauko-ohjaukseen.

LIRC ei automaattisesti tiedä minkälaisia komentoja sen pitäisi lähettää tietyille laitteelle, vaan ne täytyy nauhoittaa LIRC:in mukana

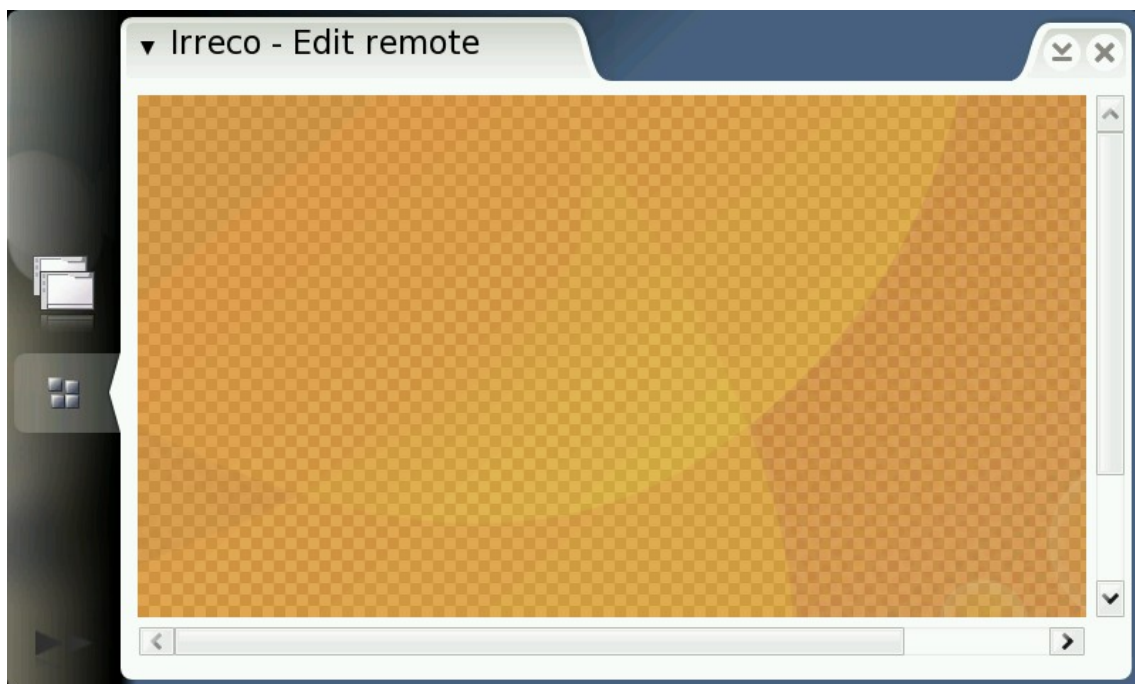
tulevaa irrecord ohjelmaa käyttäen. Kyseinen ohjelma kysyy jokaisen komennon nimen ja pyytää sen jälkeen käyttäjää tähtäämään kauko-ohjaimella IR-vastaanottimeen sekä painamaan kaukosäätimen painiketta. Tällöin irrecord nauhoittaa kaukosäätimen lähettämän komennon IR-signaalit. Lopputuloksena syntyy konfiguraatitiedosto, johon irrecord on tallentanut IR-komentojen nimet sekä niitä vastaavat IR-signaalit. Valmista konfiguraatitiedostoa voidaan tämän jälkeen käyttää LIRC:in palvelinohjelmiston kanssa.

Tämän lisäksi LIRC:in verkkosivuilta löytyy hakemisto, jossa on käyttäjien valmiiksi nauhoittamia konfiguraatitiedostoja sekalaisille laitteille ja kaukosäätimille. Valitettavasti kyseinen hakemisto ei kuitenkaan ole kovin kattava, joten käytännössä konfiguraatitiedostot joutuu yleensä nauhoittamaan itse. Myöskin konfiguraatitiedostojen laatu on vaihtelevaa, sillä käyttäjät ovat saattaneet nauhoittaa komentoja väärille nimille tai nauhoittaa pelkästään ne komennot, mitä itse ovat tarvinneet. Tämä ei ole kovin yllättävää kun ottaa huomioon, että monista kaukosäätimistä löytyy painikkeita kymmeniä kappaleita. Lopputuloksena hakemiston konfiguraatitiedostot voivat olla puutteellisia tai virheellisiä.

3 IRRECO

3.1 Käyttöliittymäeditori

Irrecossa on kaksi ensisijaista tilaa, jossa ohjelmisto toimii: editointitila ja käyttötila. Koska Irrecon perusajatuksena on antaa käyttäjälle vapaus luoda käyttöliittymä viihde-elektroniikan ohjaamista varten, on Irrecossa oltava käyttöliittymäeditori, jossa käyttäjä voi muokata ja rakentaa omaa käyttöliittymäänsä. Editorissa käyttäjä voi asettaa näytölle painikkeita, kytkeä komentoja näytöllä näkyviin painikkeisiin ja laitteen fyysisiin painikkeisiin sekä asettaa käyttöliittymän taustakuvan tai taustaväriin. Kuvassa 4 näkyy Irrecon tyhjä editointitila, johon käyttäjä siirtyy luomalla uuden käyttöliittymän.



KUVA 4. Tyhjä editointitila

Lähtökohtana Irrecon suunnittelussa on ollut se, että ohjelman lopullisesta käyttötarkoituksesta ei tiedetä muuta kuin se, että se on jonkinlainen kauko-ohjain. Sitä ohjaako käyttäjä videoita, televisiota, digiboxia, tai jotain muuta viihde-elektroniikkalaitetta, ei voida tietää

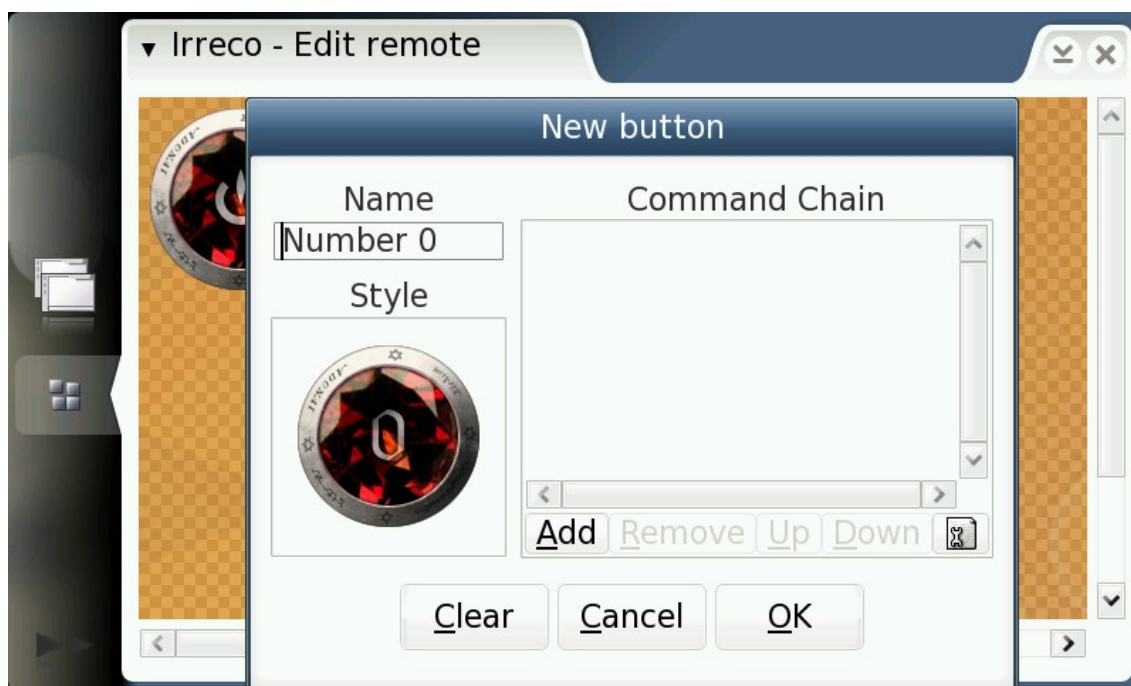
ennalta. Niinpä käyttäjä joutuu itse rakentamaan itselleen soveltuvan käyttöliittymän, eli kauko-ohjaimen, englanniksi Remote.

Irrecon lähdekoodissa Remotea kutsuttiin aluksi nimellä Layout, koska se oli ohjelmaa suunnitellessa ja toteuttaessa hyvä nimi kuvaamaan Remotea tekniseltä näkökannalta. Koska tekniseltä toteutukseltaan yksi Remote sisältää listan näppäimistä, niiden sijainneista, sekä niihin liitettyistä komennoista oli Remotea mielekästä kutsua nimellä Layout, joka englannin kielessä viittaa asioiden sijoitteluun toisiinsa nähden.

Myöhemmin kuitenkin ilmeni, että ohjelmiston käyttäjän näkökulmasta Layout ei välttämättä ole selkeä ilmaisu. Ohjelmiston käyttäjä nimittäin ajattelee käyttävänsä kauko-ohjainta, englanniksi Remote Controlleria. Tämän takia ohjelmiston käyttöliittymässä nimeksi muutettiin Remote.

3.2 Näppäimet

Käyttöliittymän muokkaus alkaa yleensä lisäämällä näytölle näppäin. Näppäintä lisättäessä käyttäjä voi asettaa näppäimelle nimen, näppäimen tyylin sekä liittää yhden tai useampia komentoja näppäimeen. Käyttäjä voi myös myöhemmin muuttaa näppäimen ominaisuuksia, jos katsoo sen tarpeelliseksi. (Kuva 5.)



KUVA 5. Ikkuna, jolla lisätään uusi näppäin

Tämän jälkeen näppäin ilmestyy näytölle, jossa käyttäjä voi vetää näppäimen haluamaansa kohtaan. Vetäminen toimii kuuntelemalla GTK+-järjestelmän lähettämiä tapahtumia, englanniksi event.

Tavanomaisessa tietokoneessa GTK+-ohjelma saa hiireltä ilmoituksia hiiren liikuttamisesta ja hiiren näppäimien painalluksista. Yleensä ohjelma saa useita ilmoituksia siitä, että hiiren kursoria siirrellään sinne tänne ruudulla, minkä jälkeen tulee yksittäisiä ilmoituksia hiiren näppäimien painalluksista. Toisin sanoen hiiren liikutus ja painikkeiden painaminen ovat toisistaan riippumattomia tapahtumia.

Internet Tablet -laiteissa "hiiren" käsittely toimii toisella tavalla, sillä laitteessa ei ole hiirtä vaan kosketusnäyttö jonka toiminta on hieman erilaista ohjelmiston näkökulmasta. Kun käyttäjä painaa kynän laitteen näyttöön kiinni saa ohjelma ensiksi ilmoituksen siitä, että "kursori" on siirretty tiettyyn sijaintiin, minkä jälkeen tulee heti ilmoitus siitä, että "hiiren" ykkösnappi on painettu alas. Laitteessa ei itsessään tietenkään ole hiirtä, tai kursoria, mutta koska GTK+-järjestelmä on alunperin suunniteltu perinteisiä tietokoneita varten, ei siinä ole varsinaista tukea kosketusnäytöille. Tämän sijaan Maemo-järjestelmää suunnitellessa on ilmeisesti nähty parhaimmaksi muuttaa kosketusnäytön tapahtumat lähimpiin vastaaviin hiiren lähettämiin ilmoituksiin. Tämän takia Maemo-ympäristössä toimiva ohjelma kuvittelee saavansa viestejä hiireltä, vaikka kyseinen hiiri onkin hieman erikoisesti käyttäytyvä hiiri.

Painikkeen vetäminen alkaa kun Irreco saa ilmoituksen, että "hiiren ensisijainen painike" on painettu alas. Tämän jälkeen jokainen "kursorin" siirto liikuttaa laitteen näytöllä näkyvää näppäintä. Painikkeen vetäminen loppuu, kun Irreco saa ilmoituksen, että "hiiren ensisijainen painike" on vapautettu. Todellisuudessa käyttäjä koskettaa kynällä laitteen näyttöä näppäimen kohdalla ja siirtää kynää näytöllä, jolloin kynän alla oleva painike seuraa kynän kosketusta näytöllä. Näin käyttäjän kannalta näppäimien asettelu laitteen näytöllä on helppoa, käyttökokemus on mukava, sekä ohjelma vastaa käyttäjän käden liikkeisiin miellyttävällä tavalla.

Yksi ongelma, mikä tuli vastaan näppäimen vetämisestä toteuttaessa, oli se, että ohjelma saa viestejä "kursorin" siirtymisestä nopeammin kuin ohjelma ehtii piirtää näytölle näppäimen uuteen paikkaan. Ratkaisuksi tähän muodostui se, että Irreco ei siirrä näppäintä heti uuteen sijaintiin, vaan tallentaa näppäimen uuden toivotun sijainnin muistiin, sekä aktivoi matalan prioriteetin funktion.

Kun näppäimen siirto on omassa funktiossa, joka suoritetaan kun

ohjelmalla ei ole muuta tehtävää, jää osa "kursorin" viesteistä huomiotta, sillä näppäimen uusi sijainti piirretään näytölle vain viimeisimmän muistiin tallennetun sijainnin perusteella. Tämä johtuu siitä, että viestejä kursorin siirtymisestä tulee useammin kuin matalan prioriteetin funktio suoritetaan. Tämä tuottaa huomattavasti mukavamman vasteen käyttäjälle kuin se, että ohjelma siirtäisi näppäintä viiveellä jokaisen sijainnin läpi, mitä käyttäjä on kynällä koskettanut. Näin tapahtuu silloin, kun ohjelma ei ehdi käsittelemään "kursorin" viestejä reaaliajassa, vaan ne jäävät jonoon ja käsitellään viiveellä.

Toisen ongelman muodosti Internet Tablet -laitteen kosketusnäytön rajallinen tarkkuus. Kun käyttäjä rakentaa omaa käyttöliittymää laitteen näytölle, olisi mielekäästä saada käyttöliittymästä siistin ja hyvin järjestellyn näköinen. Käytännössä tämä tarkoittaa näppäimien asettamista riviin näytölle. Tämä osoittautui käyttäjille vaikeaksi, sillä Internet Tablet -laitteiden kynää ja kosketusnäyttöä käytettäessä lienee käyttäjän kosketuksen tarkkuus noin 2 - 4 pikseliä suuntaan tai toiseen. Käyttäjä ei siis pysty asettelemaan näppäimiä tarkalleen haluamaansa kohtaan, johtuen kosketusnäytön ja kynän epätarkkuudesta, vaan painikkeet osuvat pikselin tai pari ohi käyttäjän tarkoittamasta kohdasta.

Ratkaisun tähän toi rakentaa Irrecon editointitilaan ristikko, jossa näppäimet osuvat ristikon koolla jaollisiin koordinaatteihin laitteen näytöllä. Käyttäjä voi asettaa ruudukon kooksi 4, 8, tai 16 pikseliä. Kun käyttäjä siirtää painiketta näytöllä, pyöristetään näppäimen koordinaatit lähimpään mahdolliseen sijaintiin ruudukossa. Näin näppäimien asettelu sujuvasti riviin tai pinoon on käyttäjälle huomattavasti helpompaa.

3.3 Fyysiset painikkeet

Koska laitteessa on joitakin fyysisiä painikkeita, olisi käyttäjälle myös kätevää, jos näihin painikkeisiin voitaisiin yhdistää komentoja viihdeelektronikkalaitteiden käskyttämistä varten. Erityisesti laitteen vasemmassa reunassa sijaitseva ristiohjain on mielenkiintoinen kauko-ohjauskäyttöä varten. Siksi ohjelmaan toteutettiin ikkuna, jossa käyttäjä voi yhdistää komentoja laitteen painikkeisiin (Kuva 6).



KUVA 6. Ikkuna, jolla yhdistetään komentoja laitteen fyysisiin painikkeisiin

3.4 Komentoketjut

Kaikki komennot, jotka Irrecossa yhdistetään joko näppäimiin tai laitteessa fyysisesti oleviin painikkeisiin, lisätään komentoketjuihin. Komentoketju on sarja komentoja, jotka suoritetaan kun painiketta painetaan. Ajatuksena komentoketjujen taustalla on käyttäjän tarve suorittaa useita komentoja yhdellä kertaa, esimerkiksi käynnistää kaksi laitetta; televisio ja DVD-soitin. Ilman komentoketjuja käyttäjä joutuisi lisäämään kaksi painiketta, yhden television käynnistämistä varten ja toisen DVD-soittimen käynnistämistä varten.

Komentoketjujen kanssa käyttäjä voi lisätä saman painikkeen taakse useita komentoja, jotka ajetaan peräjälkeen. Tällöin käyttäjä voi suorittaa monta asiaa yhdellä painalluksella, esimerkiksi käynnistää kaikki viihdelaitteet.

Toinen asia, mihin komentoketjuja voi käyttää, on valikkosurffailu. Yleensä viihdelaitteissa kaikki peruskomennot ovat suoraan kauko-ohjaimessa, kun taas vähemmän käytetyt valinnat löytyvät erilaisien valikkojen syövereistä. Käyttämällä komentoketjuja käyttäjä voi rakentaa listan komennoista, jotka avaavat valikon, siirtyvät valikoissa sopivaan kohtaan ja muuttavat asetusta halutulla tavalla.

3.5 Sisäiset komennot

Varsinaisten laitteita käskyttävien komentojen lisäksi Irreco tukee myös sisäisiä komentoja, jotka ovat ohjelman toimintaan vaikuttavia komentoja. Sisäisillä komennoilla voi muun muassa lisätä erilaisia viiveitä komentoketjujen suoritukseen tai vaihtaa aktiivista käyttöliittymää. Kuvassa 7 sisäiset komennot näkyvät "Special commands" otsikon allapuolella.



KUVA 7. Ikkuna, josta valitaan komento

Sisäiset komennot myös tarjoavat muokattavuutta Internet Tablet -laitteen fyysisten painikkeiden käyttöön. Esimerkiksi Fullscreen-painikkeen toimintaa ei ole kovakoodattu suoraan Irrecon lähdekoodeihin. Sen sijaan painikkeeseen on yhdistetty komentoketju, johon on oletuksena lisätty Irrecon sisäinen Fullscreen-komento.

Vaikka tämä voi tuntua tarpeettoman monimutkaiselta, on tämä kuitenkin kätevää, jos käyttäjä haluaa jostain syystä käyttää Fullscreen-painiketta tai mitä tahansa muuta laitteen fyysistä painiketta omien laitteidensa ohjaamiseen. Ainoan poikkeuksen näihin

muodostavat Internet Tabletin Home- ja Menu-painikkeet. Näihin painikkeisiin ei Irrecoissa voida lisätä omia komentoja, koska Internet tabletin käyttöjärjestelmä on kovakoodannut niihin omia toimintojaan, eivätkä laitteessa toimivat ohjelmat voi muuttaa niiden toimintaa.

Sisäiset komennot ovat myös hyödyllisiä, koska niitä voi lisätä komentoketjuihin kuten tavallisia komentoja. Käyttäjä voi esimerkiksi lisätä näytölle painikkeen ja lisätä Fullscreen-komennon tämän painikkeen komentoketjuun. Tämä antaa käyttäjälle suuren vapauden rakentaa käyttöliittymän toiminta halutunlaiseksi.

3.6 Teemat

Koska Irreco on ohjelmiston loppukäyttäjälle työkalu, jolla käyttäjä voi rakentaa Internet Tabletin näytölle kauniin käyttöliittymän viihde-elektroniikan ohjaamista varten, oli alusta alkaen selvää, että käyttäjät haluavat muokata ohjelman ulkoasua ja näppäimiä erinäköisiksi.

Tavallisten GTK+-näppäimien ulkoasu määritellään koko järjestelmän laajuisesti. Näin kaikkien näppäimien ulkoasu on kaikissa sovelluksissa samanlainen. Tämä on yleensä tärkeä asia rakennettaessa helppokäyttöistä tietokonejärjestelmää, sillä tällä tavoin eri ohjelmien toimintalogiikka ja käyttökokemus pysyvät samanlaisina.

Irrecon tavoitteisiin tämä ei kuitenkaan sopinut, koska tarkoituksena oli antaa käyttäjille mahdollisuus luoda uusia näppäingrafiikoita ja mahdollistaa erinäköisten näppäimien lisääminen näytölle. Siksi tavalliset GTK+-näppäimet eivät olleet kelpollisia tähän projektiin.

Ratkaisuksi muodostui simuloida oikeita näppäimiä kuvatiedostojen avulla. Irrecon näppäin koostuu kahdesta kuvasta. Yksi kuva näytetään, kun näppäin on lepotilassa, eikä sitä paineta. Toista kuvaa näytetään, kun näppäin on aktiivinen, kun sitä painetaan. Tämän

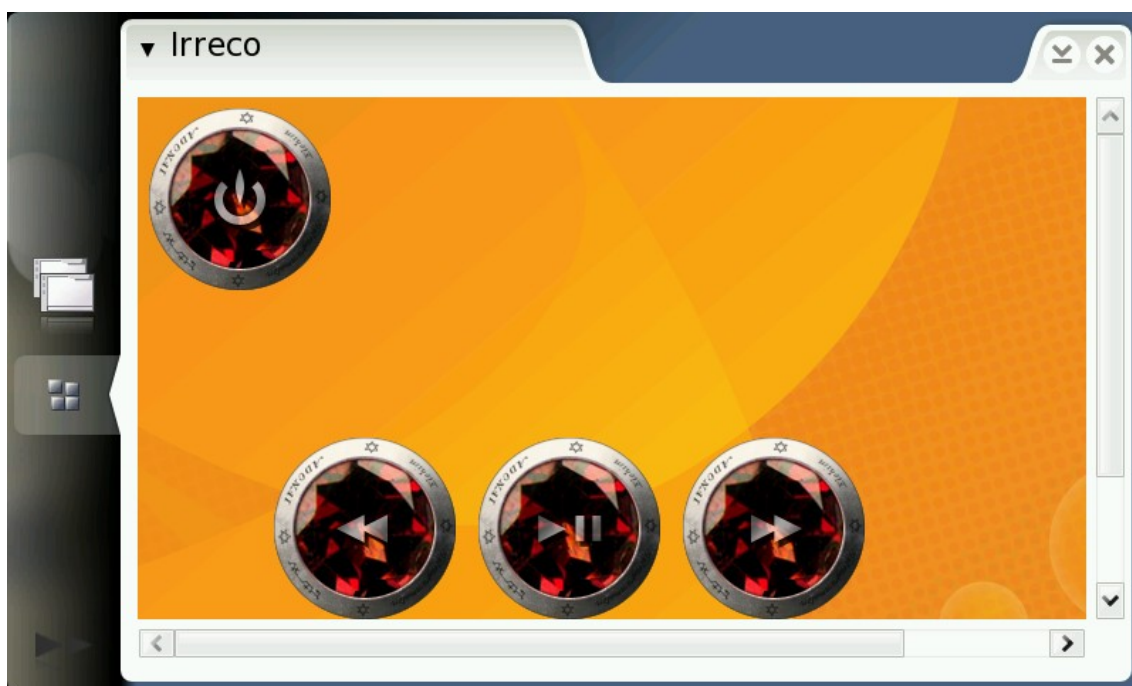
lisäksi Irreco piirtää kuvien päälle otsikkotekstin, jos käyttäjä haluaa syöttää sellaisen.

Tämä järjestelmä myös antaa käyttäjälle mahdollisuuden luoda uusia näppäintyylejä suhteellisen helposti, sillä näppäintä varten vaaditaan vain kaksi kuvatiedostoa.

Projektin aikana oli mahdollista käyttää Nokian resursseja teemojen tuottamiseen, joten niitä teetettiin neljä kappaletta. Näihin teemoihin tarvittavien muutamien kuvatiedostojen tuottaminen ei ollut Nokian kannalta iso panostus projektiin, mutta Irrecon kannalta ne tekivät suuren parannuksen ohjelmiston ilmeeseen ja käytettävyyteen.

3.7 Käyttötila

Käyttötilassa Irreco näyttää käyttäjän editointitilassa luomia käyttöliittymiä ja odottaa, että käyttäjä painaa jotakin ruudulla näkyvää näppäintä tai laitteen fyysistä painiketta. Kun käyttäjä painaa jotakin painiketta, suorittaa Irreco siihen yhdistetyn komentoketjun komennot. Jos käyttäjä on luonut useita käyttöliittymiä, on aktiivista käyttöliittymää mahdollista vaihtaa joko Irrecon valikosta tai painamalla näppäintä, jonka komentoketjuun on yhdistetty komento vaihtaa käyttöliittymää. (Kuva 8.)



KUVA 8. Irreco ikkunoidussa käyttötilassa

Käyttötilaa on myös mahdollista käyttää koko ruudun täyttävässä Fullscreen-tilassa, jolloin Maemo-käyttöjärjestelmän valikot ja otsikkopalkki piilotetaan, ja Irreco ottaa käyttöönsä koko ruudun (Kuva 9). Käytännössä Irrecoa käytetään varsinaiseen kauko-ohjaukseen yleisimmin juuri Fullscreen-tilassa siten, että käyttäjän rakentama kauko-ohjain täyttää koko ruudun. Laitteen käyttömukavuuden kannalta on kuitenkin mielekkäämpää, että Irreco käynnistyy ikkunoituun tilaan sen sijaan, että valikkopalkit katoaisivat näkyvistä. Näin käyttäjä voi helposti sulkea Irrecon, vaihtaa aktiivista käyttöliittymää tai vaikka siirtyä editointitilaan suoraan Maemon käyttöliittymän valintojen avulla.



KUVA 9. Irreco kokoruudun käyttötilassa

Loppujenlopuksi käyttötila on kuitenkin melko yksinkertainen tila, jossa käyttäjän tekemä työ realisoituu käytettäväksi kauko-ohjaimeksi.

4 KOMENTOJÄRJESTELMÄT

Tässä opinnäytetyössä komentojärjestelmällä tarkoitetaan järjestelmää, jolla voidaan kontrolloida muita laitteita tai yksittäistä laitetta, jota voidaan kontrolloida viestimällä sille jollakin tietyllä tavalla. Käytännössä hyvin monia tietoteknisiä laitteita ja ohjelmistoja voidaan kauko-ohjata tai etähallita, jos niille viestitään oikealla tavalla. Valitettavasti vain nämä tavat voivat olla hyvinkin erilaisia, joten yksittäisen toteutuksen rakentaminen Irrecoon, jolla voisi kontrolloida kaikkia laitteita, ei ole mahdollista.

Erilaisia komentojärjestelmiä ovat esimerkiksi LIRC, IRTrans ja MythTV. LIRC ja IRTrans ovat molemmat komentojärjestelmiä, joilla voidaan kontrolloida viihde-elektroniikkaa IR-lähettimien avulla. Periaatteessa molemmat järjestelmät siis toimivat samalla tavalla ja toteuttavat saman tehtävän. Silti kumpikin järjestelmä sisältää oman TCP/IP pohjaisen verkkoprotokollan, mitä käyttäen ohjelmistojen tulee viestiä niille. MythTV taas on Linux-tietokoneissa toimiva kotiteatteriohjelmisto, jolla käyttäjä voi katsoa esimerkiksi televisiota tai videoita. Ja kuten tavallista, MythTV sisältää tietenkin oman protokollan, jonka avulla sitä voidaan etähallita.

Koska Irrecon kuitenkin tuli tukea useita erilaisia komentojärjestelmiä, piti ohjelmaan suunnitella jonkinlainen järjestelmä, jonka avulla Irreco voi kommunikoida useiden komentojärjestelmien kanssa yhtä aikaa. Toisaalta yksittäiset komentojärjestelmät tulisi paketoita omiksi kokonaisuuksikseen siten, että uuden komentojärjestelmän lisääminen ei vaikuttaisi toisten komentojärjestelmien toimintaan.

Tällainen paketointi voidaan saavuttaa monin eri tavoin, kuitenkin järkevintä on yleensä käyttää sitä mekanismia, jota kukin ohjelmointikieli tai kehitysympäristö tukee luontaisesti. C-ohjelmoinnissa tämä mekanismi on tehdä kirjastoja. Tämän lisäksi GLib-kirjasto sisältää GModule-järjestelmän, jonka avulla

tämänkaltaisista apukirjastoista voidaan rakentaa helposti käytettäviä moduuleja GLib-ympäristöön.

Näinpä ratkaisuksi muodostui rakentaa jokaisen komentojärjestelmän yksityiskohtien hallinta omaksi kirjastoksi, käyttää kirjastojen käsittelyyn GModule-järjestelmää, ja rakentaa Irrecon ja näiden kirjastojen välille rajapinta. Näin toteutettuna Irrecon ei tarvitse ymmärtää tietyn komentojärjestelmän yksityiskohtia vaan Irreco voi pyytää kirjastolta listan komennoista, joita kukin komentojärjestelmän avulla hallittava laite tukee. Tämän jälkeen Irreco voi lähettää jonkin komennon toteutettavaksi kirjastolle, joka taas välittää sen eteenpäin komentojärjestelmälle oikealla tavoin kommunikoiduna. Lähdekoodissa näitä kirjastoja kutsutaan nimellä backend.

4.1 Backend-kirjastojen toteutuksesta

Se, että komennot, joita Irreco voi suorittaa, tulevat backend-kirjastoilta, jotka lukevat ne joistakin tuntemattomista järjestelmistä, aiheuttaa ongelman. Irreco ei voi luottaa siihen, että komennot jotka olivat vielä hetki sitten käytettävissä, ovat yhä käytettävissä. Verkkoliikenne saattaa katketa, käyttäjä saattaa muuttaa backendin asetuksia, palvelin toisessa päässä saattaa kaatua, käyttäjä sammuttaa tietokoneensa, jne. Mikä tahansa näistä muutoksista voi sattua kun Irrecoa käytetään tai Irrecon ollessa suljettuna.

Tämän takia Irreco pyrkii parhaansa mukaan tarkistamaan, että kaikki komennot, jotka on lisätty komentoketjuun, ovat yhä käytettävissä, kun jotain näppäintä painetaan, mutta käytännössä Irreco ei voi virhetilanteen sattuessa tehdä muuta kuin näyttää virheilmoituksen.

Nykyään kodeista löytyy viihde-elektroniikkalaitteita useista huoneista. Jos käyttäjä esimerkiksi haluaisi ohjata laitteita kahdessa eri huoneessa, voisi hänellä olla sitä varten yksi LIRC:llä varustettu tietokone jokaista huonetta varten. Irrecon näkökulmasta tämä tarkoittaa sitä, että yhdestä backendistä voi olla käytössä useita

kopioita, eli instansseja. Tässä esimerkissä LIRC backendistä olisi käytössä kaksi instanssia, yksi jokaista huonetta ja tietokonetta kohden.

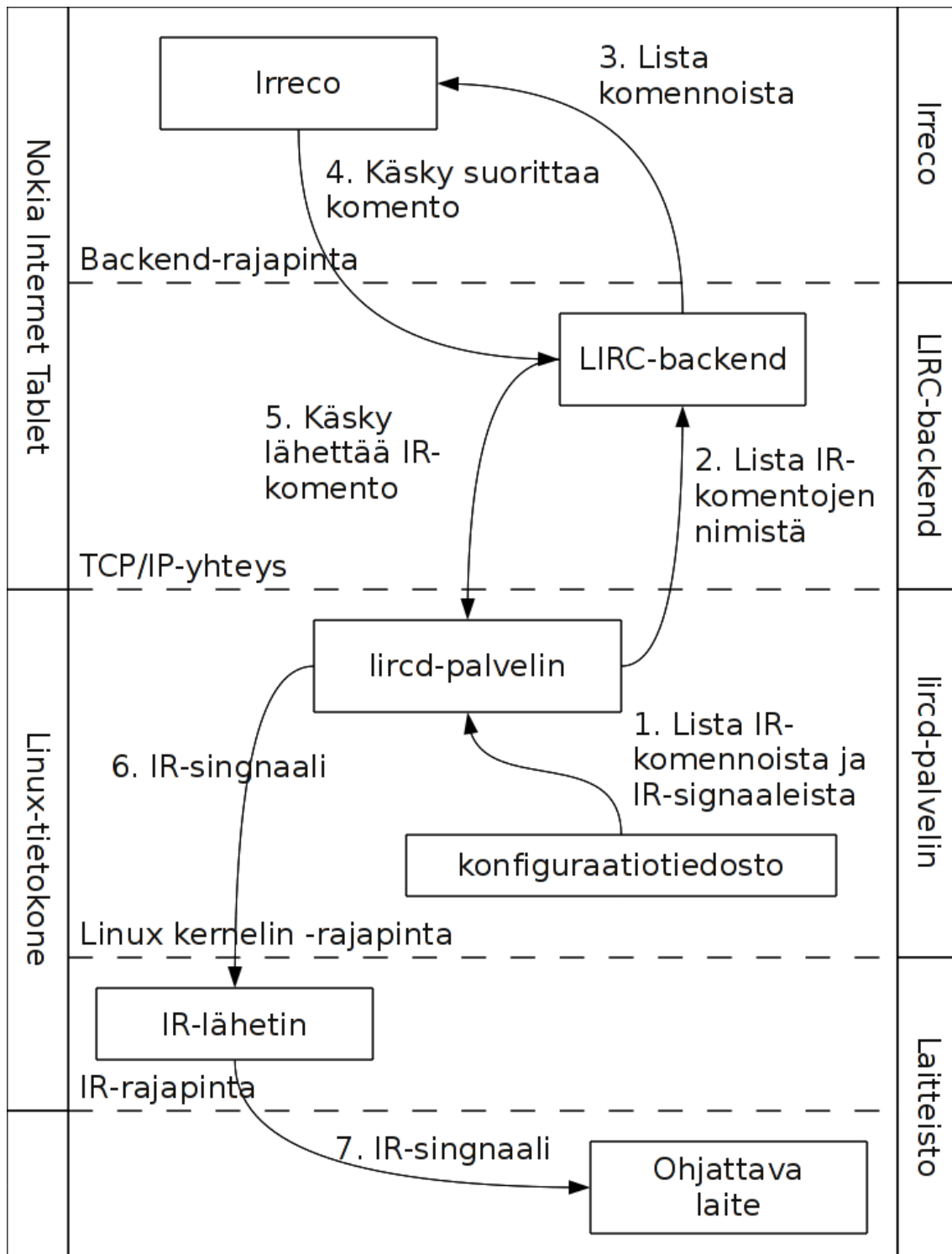
Vaikka laajennettavan backend-järjestelmän rakentaminen mahdollistaa useiden komentojärjestelmien tukemisen, se myös monimutkaistaa komentojen suorittamista. Irreco ei voi suoraan lähettää televisiolle komentoa vaihtaa kanavaa, vaan joutuu sen sijaan ensin selvittämään oikean backendin ja instanssin mistä komento on listattu. Tätä varten Irreco joutuu ylläpitämään tietorakenteita joihin on listattu backendien, instanssien, laitteiden ja komentojen väliset suhteet.

Myös backend-kirjastojen tekninen toteutus tuo omat haasteensa, sillä ne ovat C:llä ohjelmoituja kirjastoja, jotka suorittamista varten ladataan Irrecon muistiavaruuteen. Tämä tarkoittaa sitä, että backendissä mahdollisesti oleva virhe voi kaataa koko Irrecon, koska suorituksen aikana Irreco ja backendit ovat yhtä ja samaa kokonaisuutta.

Tämän lisäksi Irrecon pitää ajon aikana tarkistaa, että backend on rekisteröinyt kaikki vaaditut funktiot. Olemattoman funktion kutsuminen voisi nimittäin myöskin kaataa Irrecon. Lopputuloksena backend-järjestelmä mahdollistaa Irrecon laajentamisen, mutta toisaalta monimutkaistaa ohjelmiston toimintaa ja toteutusta.

4.2 LIRC-backend

LIRC-backend on Irrecon backendi, eli laajennuskirjasto, jonka avulla Irreco kykenee viestimään LIRC-ohjelmiston lircd-palvelimelle. Lircd-palvelimen toiminta perustuu palvelin - asiakas -malliin, jossa palvelin odottaa asiakkaan yhteydenottoa ja asiakkaan lähettämiä käskyjä. Kun palvelin vastaanottaa asiakkaan, eli Irrecon, lähettämän komennon, se lähettää komentoa vastaavan IR-signaalin Linux-tietokoneeseen kiinnitetyn IR-lähettimen kautta.



KUVA 10. Irrecon, LIRC-palvelimen ja LIRC-backendin yhteistoiminta

Kuvassa 10 esitetään kuinka Irreco, LIRC-palvelin ja LIRC-backend toimivat ja viestivät toisilleen. Tarkemmin selvitettynä kuvassa tapahtuu seuraavaa:

1. Linux-tietokoneeseen asennettu lircd-palvelin lukee /etc/lirc/lircd.conf tiedoston. Tämä tiedosto sisältää listan laitteista, IR-signaaleista ja komentojen nimet.
2. Irreco pyytää LIRC-backendia listaamaan käytettävissä olevat laitteet ja komennot. Tätä varten LIRC-backend ottaa TCP/IP-yhteyden lircd-palvelimeen, jolta LIRC-backend kysyy listan laitteista ja komennoista, joita voidaan käskyttää kyseisen palvelimen avulla.
3. LIRC-backend välittää listan eteenpäin Irrecolle. Olennaisinta tässä vaiheessa on se, että lircd-palvelin ja Irreco käsittelevät listaa eri muodoissa. Näin LIRC-backendin päätehtäväksi jää hakea komentolista palvelimelta ja muuttaa se Irrecon ymmärtämään muotoon.
4. Käyttäjä luo näppäimen, johon on kytketty jokin LIRC-backendin listaamista komennoista. Kun käyttäjä painaa kyseistä näppäintä käyttötilassa, lähetetään komento LIRC-backendille suoritettavaksi.
5. LIRC-backend ottaa TCP/IP-yhteyden lircd-palvelimeen ja kääntää palvelinta suorittamaan komennon.
6. lircd-palvelin etsii komennon nimeä vastaavan IR-signaalin /etc/lirc/lircd.conf tiedostosta luettujen tietojen perusteella ja lähettää IR-signaalin eteenpäin IR-lähettimen kautta.
7. Jokin laite vastaanottaa IR-signaalin.

5 POHDINTA

Opinnäytetyötä tehdessä ongelmana oli se, että alun pitäen työtä pidettiin Nokian puolelta eräänlaisena harjoitteluprojektina tai prototyypinä, jossa voitaisiin kouluttaa uusia ohjelmoijia Maemo-sovelluskehitystä varten sekä samaan aikaan tuottaa mielenkiintoinen apuohjelma Maemo-ympäristöön. Tästä johtuen ohjelmistolla ei varsinaisesti ollut tarkkaan määritettyä aikataulua eikä tarkkaan määriteltyä vaatimusmäärittelyä. Ajan kuluessa tämä taas johti siihen, että niin minä kuin myöskin Nokian esimieheni, keksimme uusia mielenkiintoisia ominaisuuksia, joita ohjelmistoon olisi kiva toteuttaa. Tämä johti niin sanottuun "feature creep" ongelmaan, eli siihen että ohjelmistoon ajan mittaan hiipii yhä uusia ominaisuuksia ja työn laajuus kasvaa. Tämä myös kasvattaa lähdekoodin laajuutta ja tekee ohjelmiston kehittämisestä vaikeampaa ja hitaampaa. Ohjelmistosta tulee myös monimutkaisempi käyttää, kun siinä on paljon ominaisuuksia.

Opinnäytetyön yhtenä päämääränä oli kehittää Irrecosta avoimen lähdekoodin projektia ja yrittää rakentaa yhteisöä Irrecon ympärille. Tämä osoittautui vaikeaksi. Yhteisön rakentaminen vaatisi sen, että ihmisiä saadaan mukaan projektiin, mutta se ei ole helppoa. Ihmisistä ohjelma voi olla "ihan kiva", mutta se ei tarkoita sitä, että he olisivat valmiita tekemään työtä sen parantamiseksi. Myöskin käyttäjien ja ohjelmistokehittäjien ero on ymmärrettävä. Käyttäjät haluavat helpon sovelluksen, kehittäjät taas haluavat helpon pääsyn ohjelmiston lähdekoodeihin, teknisiin yksityiskohtiin ja kehitystyökaluihin. Jos nämä tekniset asiat ovat näkyvästi esillä verkossa, saattavat käyttäjät säikähtää. Jos ne taas eivät ole näkyvillä, eivät teknisistä asioista kiinnostuneet ihmiset välttämättä innostu projektista. Vaikka ohjelmistolla olisikin paljon käyttäjiä, se ei silti välttämättä saa paljon kehittäjiä mukaansa tai toisinpäin. Monet isommat avoimen lähdekoodin projektit ovat ratkaisseet tämän tekemällä verkkosivuille

erilliset alueet käyttäjille ja ohjelmistokehittäjille.

Yhteydenpitoa varten Irrecolle tehtiin sähköpostilista Maemo Garageen ja keskustelu Internet Tablet Talk -sivuston foorumiin. Linkki molempiin laitettiin Irrecon verkkosivuille. Sähköpostilistalle ei tullut yhtään viestiä. Keskusteluun taas tuli kymmeniä viestejä ohjelmiston käyttäjiltä. Tästä voi tulkita, että nykyajan nettikäyttäjille on helpompaa viestiä foorumien kautta, kuin lähettää viestejä sähköpostilistoille.

Erityisesti avoimen lähdekoodin maailmassa olisi hyvin tärkeää, että kaikki ohjelmistoon liittyvä kommunikaatio ja dokumentaatio olisi saatavilla internetissä ja myös käytäisiin internetissä. Nokialla tämä osoittautui ongelmaksi, sillä tein työtä Nokialla ja julkaisin tuotokseni silloin tällöin. Tulkitsen sähköpostilistan epäonnistumisen tämän työtavan syyksi. Yleensä sähköpostilistoilla käydään juuri ohjelmiston kehityskeskusteluja ohjelmistokehittäjien välillä. Koska en itse käyttänyt sähköpostilistaa mihinkään, ei kukaan muukaan lähettänyt sinne mitään.

Myöskin versionhallintaohjelmisto osoittautui haastavaksi. Käytin alunperin SVN-versionhallintaohjelmistoa, mutta julkaisua varten lähdekoodi tuli saattaa Maemo Garageen saataville. Tämä osoittautui vaikeaksi, sillä tein kehitystyötä omalla kannettavallani ja tallensin muutoksen kannettavassa tietokoneessa sijainneeseen SVN-repositoryyn. Valitettavasti SVN ei tarjoa helppoa tapaa synkronoida tehtyä työtä yhdestä repositorystä toiseen. Minkä takia lähdekoodien kopiointi Maemo Garagen SVN-repositoryyn oli tarpeettoman hankalaa. Näin jälkeempäin katsottuna kaikki ohjelmointityö olisi tullut tehdä suoraan Maemo Garagen repositoryyn tai käyttäen esimerkiksi Git-versionhallintaohjelmistoa, jossa kahden repositoryn synkronoiminen on helppoa.

Ohjelmointiympäristönä GTK+ on vaikea. Ongelmana on se, että

GTK+ on rakennettu C-kielellä ohjelmoidun olio-ohjelmointi ominaisuudet tarjoava GObject-järjestelmän avulla. Vaikka tällä tavoin voikin tehdä olio-ohjelmointia C-kielellä, on se silti hyvin kankeaa ja hidasta. Suuri osa lähdekoodista on erilaista "peruskauraa", joka muissa erityisesti olio-ohjelmointiin suunnitelluissa ohjelmointikielissä tapahtuisi automaattisesti taustalla. Tästä syystä jonkin periaatteessa hyvinkin yksinkertaisen luokan ohjelmoiminen GTK+- ja GObject-ympäristössä voi viedä satoja rivejä koodia, jotta asiaankuuluvat funktiot ja ominaisuudet saadaan rekisteröityä GObject-järjestelmään.

Toisaalta GObject-järjestelmä tuo myös tiettyjä etuja. Koska C-kääntäjiä on olemassa käytännössä kaikille laitearkkitehtuureille, on GObject-lähdekoodi mahdollista kääntää ja suorittaa useissa ympäristöissä. Monet ohjelmointikieliset tai niiden kääntäjät on myös itsessään ohjelmoitu C-ohjelmointikielillä. Siksi GTK+-järjestelmän integrointi muihin ohjelmointiympäristöihin on mahdollista. GTK+-kirjasto voikin käsittääkseni käyttää ainakin noin 20:stä muusta ohjelmointiympäristöstä ja ohjelmointikielestä käsin.

Jos tulevaisuudessa joudun ohjelmoimaan GTK+-ohjelmistoja, pitäisin järkevänä kokeilla jotain näistä muista 20:stä ohjelmointiympäristöstä. Erityisesti Python ja Vala vaikuttavat mielenkiintoisilta. Tällä tavoin voisin ohjelmoijana keskittyä ohjelmiston rakentamiseen, GTK+-kirjaston huolehtiessa alemmalla tasolla käyttöliittymän piirtämisestä ja muista nopeuskriittisistä asioista. Ohjelmoijana tällainen ratkaisu, jossa yhdistyy korkean tason ohjelmointikielen helppokäyttöisyys ja matalantason ohjelmointikielillä ohjelmoidun GTK+-kirjaston tehokkuus, on mielenkiintoinen.

Kokonaisuutena opinnäytetyöprojekti onnistui ja Irreco yleensä ottaen täyttää sille asetetut vaatimukset ja toiveet. Toisaalta opinnäytetyötä tehdessä ilmeni monta asiaa, mitkä olisi voinut toteuttaa tai tehdä toisinkin.

LÄHTEET

About Debian, Debian.org. Hakupäivä 31.3.2011.

<http://www.debian.org/intro/about>.

All Debian Packages in "squeeze", Debian.org. Hakupäivä 31.3.2011.

<http://packages.debian.org/stable/allpackages?format=tgtz>.

GTK+ Overview, Gtk.org. Hakupäivä 25.4.2011.

<http://www.gtk.org/overview.html>.

Maemo Diablo Reference Manual for maemo 4.1, Maemo.org.

Hakupäivä 31.3.2011.

http://maemo.org/maemo_release_documentation/maemo4.1.x/Maemo_Diablo_Reference_Manual_for_maemo_4.1.pdf.

Moody, Glyn. 2001. Kapinakoodi: Linus Torvalds ja vapaan ohjelmoinnin vallankumous. Suom; Riikka Toivanen, Heikki Karjalainen.

Search, Sourceforge.net. Hakupäivä 18.3.2011.

<http://sourceforge.net/search/>.

Search Distributions, DistroWatch.com. Hakupäivä 31.3.2011.

<http://distrowatch.com/search.php>.

Services & Products, RedHat.com. Hakupäivä 31.3.2011.

<http://www.redhat.com/products/>.

Ubuntu for your business, Ubuntu.com. Hakupäivä 31.3.2011.

<http://www.ubuntu.com/business>.