Qt –pohjaisen paikkatietotyökalun suunnittelu ja toteutus

(Julkinen osa)

Samu Laaksonen

Opinnäytetyö
Huhtikuu 2012
Tietotekniikan koulutusohjelma
Ohjelmistotekniikka

TAMPEREEN AMMATTIKORKEAKOULU
Tampere University of Applied Sciences

TIIVISTELMÄ

Tampereen ammattikorkeakoulu
Tietotekniikan koulutusohjelma
Ohjelmistotekniikka

LAAKSONEN SAMU:
Qt -pohjaisen paikkatietotyökalun suunnittelu ja toteutus (Julkinen osa)
Opinnäytetyö 16 sivua
Helmikuu 2012

Työn ohjaaja: Tony Torp

Tässä työssä käsitellään Qt –sovelluskehykseen perustuvan paikkatietotyökalun suunnittelua ja toteuttamista.

Tässä työssä katetaan Qt –sovelluskehyksen perusteita ja periaatteeita. Samalla otetaan kantaa kyseisen sovelluskehyksen soveltuvuuteen työpöytäsovelluksia kehitettäessä.

Työssä käsiteltävä sovellus toteutettiin päivitystarpeena korvaamaan sitä edeltänyttä, vanhemmilla tekniikoilla toteutettua työpöytäsovellusta. Samalla sovellukseen lisättiin uusia ominaisuuksia.

Työn sisällön avulla työn lukija saa toivottavasti ideoita ja ajateltavaa koskien varsinkin työpöytäsovellusten käyttöliittymien suunnittelemista ja toteutusta.

Työn loppupuolella käydään läpi sovelluksen integrointia osaksi suurempaa järjestelmää.

Työ sisältää toteutetun sovelluksen rakenteen ja siihen liittyvän tutkimustyön suhteen luottamuksellista tietoa, jotka on poistettu julkisesta versiosta.

Asiasanat: Qt, paikkatieto, käyttöliittymä

ABSTRACT

Tampereen ammattikorkeakoulu
Tampere University of Applied Sciences
Degree Programme in Information Technology
Option of Software Engineering

LAAKSONEN SAMU:
Planning and Implementation of Qt-based Location Data Application (Public part)
Bachelor's Thesis 16 pages
April 2012

Thesis supervisor: Tony Torp

This thesis covers designing and development of a Qt framework based desktop application that uses location based data.

This thesis also covers basics and principles of the Qt framework. At the same time points are taken regarding Qt frameworks suitability on general desktop application development.

Application in question was developed as an upgrade to replace an older version, which had been developed using older technologies. Compared to old version, application also gained some new features.

By reading this thesis, the reader hopefully gains some ideas and information regarding graphical user interfaces on desktop applications. Designing and implementing of the graphical user interface happened to be a major part of the application in question.

End part of the thesis consists of integration of the application to be part of a bigger system.

This thesis contains confidential information regarding the structure of the software and research relating to it. In question information has been removed from the publicly available version.

Keywords: Qt, location based data, graphical user interface

ALKUSANAT

Työssä käsittelen Qt –ohjelmistokehykseen perustuvan paikkatietosovelluksen suunnittelemista ja toteutusta.

Sovelluksen toteutuksessa käytetyn iteratiivisen mallin ansiosta pääsin jatkuvasti parantamaan sovelluksen rakennetta. Uskonkin tämänlaisen työskentelymallin olevan yksi suurimmista tekijöistä projektin onnistumiseksi.

Työn ansiosta opin paljon työpöytäsovellusten ja varsinkin käyttöliittymien suunnittelemisesta ja toteutuksesta.

Kiitokseni haluan osoittaa työn tilaajalle.

Tampereella 12.4.2012

_____

Samu Laaksonen

SISÄLLYS

(Sovelluksen toteutus- ja suunnitteluosiot poistettu luottamuksellisuuden vuoksi)

TERMIT JA LYHENTEET

AdoDB	Tietokannan abstraktointi kerros, mahdollistaa sovellusten kehittämisen ilman, että kehitettävään sovellukseen tulee riippuvuutta yhdestä ja tietystä tietokannasta
C++	Ohjelmointikieli
GNU	Projektin nimi, jonka tavoitteena on kehittää täysin vapaa käyttöjärjestelmä
IDE	Integrated Development Environment, integroitu kehitysympäristö, erillinen sovellus jonka avulla tiettyyn sovelluskehykseen perustuvia sovelluksia on helpompi kehittää
moc	Qt:n käyttämä meta-object compiler järjestelmä
Qt	Qt –sovelluskehys, tarjoaa valmiita komponentteja c++ ohjelmointiin
QtSDK	Qt Software Development Kit, Qt –sovelluskehystä käyttävien sovellusten kehittämiseen tarkoitettu kokonaisuus
STL	Standard Template Library, C++ ohjelmointikielen kirjastolaajennus
SQL	Structured Query Language, standardoitu kyselykieli, jolla relaatiotietokantaan voi tehdä erilaisia hakuja
TCP/IP	Transmission Control Protocol / Internet Protocol, usean Internet liikennöinnissä käytettävän tietoverkkoprotokollan yhdistelmä
www	World Wide Web, Internet-verkossa toimiva hajautettu hypertekstijärjestelmä

1 JOHDANTO

Tämä työ käsittelee Qt –ohjelmistokehykseen pohjautuvaa, paikkatietoa hyödyntävää työpöytäsovellusta. Sovelluksen käyttöliittymä ja logiikka on ohjelmoitu käyttäen C++ ohjelmointikieleen perustuvaa Qt –sovelluskehystä.

Opinnäytetyön alkuperäisen version johdanto sisältää luottamuksellista tietoa, ja siitä syystä johdantoa on lyhennetty julkista versiota varten.

Työssä käsiteltävään sovellukseen liittyvät suunnittelua ja toteutusta koskevat kappaleet on luokiteltu luottamukselliseksi tiedoksi, ja ne on täten poistettu julkisesti jaossa olevasta versiosta. Näihin lukeutuvat kappaleet 2, 4 ja 5.

3 QT FRAMEWORK

3.1 Qt

Qt-framework, eli Qt –ohjelmistokehys on alustariippumaton ohjelmistojen kehitysympäristö. Qt sisältää sekä graafisten käyttöliittymien kehittämiseen tarvittavan kehitysympäristön, että standardin C++ luokkakirjaston. Qt on olio pohjainen, event-handler mallia toteuttava ohjelmistokehys.

Alkujaan Qt on lähtöisin Norjalaisen Trolltech yhtiön projektina. Vuonna 2008 Nokia Oyj kuitenkin osti Trolltechin ja nimesi sen ensin Qt Softwareksi, ja myöhemmin Qt Development Frameworks:ksi.

Qt:n ollessa alustariippumaton kehitysympäristö, tarkoittaa se sitä, että sillä pystyy kehittämään sekä Windows, Linux että Mac OS X ympäristöissä samanaikaisesti toimivia ohjelmistoja. Lisäksi Nokia on tehnyt Qt:sta versiot myös omille Symbian ja Meego käyttöjärjestelmilleen, täten sitä on mahdollista käyttää myös mobiiliohjelmoimiseen.

3.1.1 Meta-object compiler

Qt tarjoaa ohjelmistokehittäjän käytettäväksi muutamia siihen sisäänrakennettuja ominaisuuksia, joilla tähdätään helpottamaan ja nopeuttamaan ohjelmiston kehittämistä.

Keskeisimpänä Qt:n tarjoamana ominaisuutena on meta-object compiler, eli moc –systeemi. Moc tarjoaa mahdollisuuden käyttää signals & slots (signaalit & slotit) ominaisuutta ohjelmiston sisäisessä kommunikaatiossa eri objektien välillä, ajonaikaista tyyppi-informaatiota objekteista sekä lisää dynaamisen ominaisuusjärjestelmän objekteille.

Näistä ominaisuuksista signals & slots on selkeästi käytetyin ja siihen perehdymmekin tarkemmin kappaleessa 3.1.2.

Ottaakseen moc –systeemin käyttöön ohjelmassaan on kehittäjän huolehdittava kolmesta asiasta. Ensinnäkin moc –systeemi on käytettävissä vain luokissa jotka perivät QObject:n. Tämä ehto täyttyy yleisessä Qt ohjelmoinnissa helposti, koska lähes kaikki Qt:n omat luokat perivät joko QObject:n suoraan, tai sitten luokan joka on perinyt QObject:n.

Toiseksi luokkaan, jonka yhteydessä halutaan käyttää moc –systeemiä tulee lisätä Q_OBJECT macro. Moc kääntäjä lukee C++ lähdetiedoston ja löytäessään luokkaesittelyn, jonka yhteydessä on mainittu Q_OBJECT macro, se tuottaa automaattisesti erillisen C++ lähdekooditiedoston (moc_*.cpp), joka sisältää luokan meta-object ominaisuudet. Nämä moc_*.cpp tiedostot käännetään .o object tiedostoiksi ennen ohjelman kääntämisen linkitysvaihetta, ja täten lisätään mukaan suoritettavaan ohjelmaan.

Kolmantena huomiona kehittäjän on varmistuttava, että ohjelman käännösvaiheessa erillinen moc –työkalu tulee kutsutuksi. Nokian tarjoamalla QtSDK :lla ohjelmoitaessa tämä vaihe on automaattinen, koska käännöstyökalut kutsuvat qmake työkalua, johon moc –työkalu on automatisoitu. Mutta esimerkiksi käytettäessä GNU make työkalua tulee kehittäjän itse lisätä moc –tiedoston generoiva kutsu käännettävän projektin make tiedostoon.

3.1.2 Signals & Slots

Signals & Slots järjestelmä pohjautuu Qt:n tarjoaman meta-object compilerin luomiin moc_*.cpp tiedostoihin. Signaaleja ja slotteja käytetään ohjelman sisäiseen kommunikaation eri objektien välillä. Tämä ominaisuus lienee Qt:n suurin eroavaisuus verrattaessa toisiin ohjelmistokehyksiin.

Tarve signaaleille ja sloteille perustuu ajatukseen, että ohjelmoitaessa graafisella käyttöliittymällä toimivaa sovellusta, on usein tarpeellista tiedottaa käyttöliittymän eri

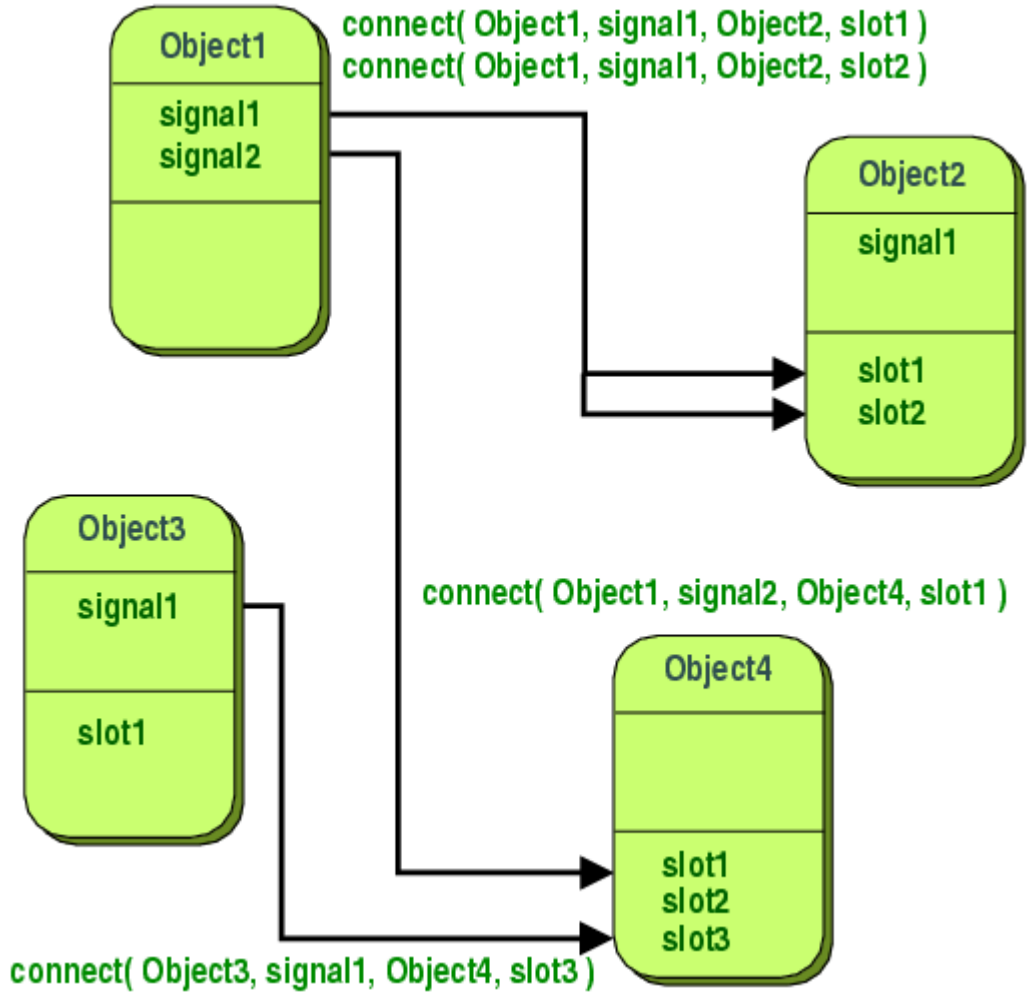
komponentteja toisten komponenttien tilanmuutoksista. Hyvänä esimerkkinä tästä voisi käyttää numeronäyttöä, joka näyttää liukusäätimen arvon.

Monissa vanhemmissa ohjelmistokehyksissä tämänkaltainen kommunikaatio eri objektien välillä on usein hoidettu käyttämällä takaisinkutsu rakennetta. Takaisinkutsujen käyttämisessä on kuitenkin kaksi perustavanlaatuista ongelmaa. Ensinnäkin, ne eivät ole turvallisia takaisin kutsuun käytettyjen argumenttien suhteen. Toisekseen, takaisinkutsujen käyttäminen luo vahvan sidoksen kutsuvan ja kutsuttavan funktion välille, tämä johtaa turhiin riippuvuussuhteisiin ohjelmiston rakenteessa.

Qt:n signals & slots järjestelmä tarjoaa kätevän vaihtoehdon perinteiselle takaisinkutsu rakenteelle. Lähes kaikki Qt:n mukana tulevat komponentit tarjoavat ohjelmoijan käyttöön valmiita signaaleja ja slotteja käytettäväksi käyttöliittymäohjelmoinnissa. Yleinen käytäntö kuitenkin on, että monet ohjelmistokehystä käyttävät ohjelmoijat kirjottavat omat slot metodinsa, saaden näin täyden hallinnan tapahtumaketjusta.

Signaalien ja slottien käyttäminen Qt –pohjaisessa ohjelmistossa onkin suhteellisen yksinkertaista. Koska signals & slots järjestelmä tukeutuu Qt:n meta-object compileriin, tarvitsee sitä käyttävässä ohjelmassa huolehtia, että luvussa 3.1.1 mainitut ehdot täyttyvät.

Signaaleja ja slotteja on mahdollista hyödyntää monipuolisesti, yhden signaalin voi kytkeä moneen slottiin ja yhteen slottiin on mahdollista kytkeä monia signaaleja. Alla olevassa kuviossa on esitetty osa järjestelmän käyttömahdollisuuksista.



KUVIO 3: Esimerkkikuva signals & slots järjestelmän käytöstä (Qt 4.8: Signals & Slots).

Seuraavassa on esimerkkikoodilistaus signals & slots järjestelmän käytöstä oikeassa sovelluksessa.

```
QAction* closeButton = m_mainMenu->addAction("Close");
connect(closeButton, SIGNAL(triggered()), this, SLOT(close()));
```

Koodiesimerkki 1: Signals & slots connect

Tässä esimerkissä luodaan ensin QAction, joka asetetaan m_mainMenu nimiseen QMenu:un. QMenu toteuttaa sovellukseen valikko ominaisuuden. Kuten ensimmäisestä rivistä käy ilmi, annetaan QActionin loppukäyttäjälle päin näkyväksi nimeksi ”Close”, tätä valintaa painamalla käyttäjän on siis tarkoitus saada sovellus suljettua. Toisella rivillä jokaisen QActionin valmiina tarjoama triggered() signaali yhdistetään kuvitteellisen QWidgetin this pointerin kautta close() –slottiin. Close() –slot on jokaisen QWidgetistä perivän luokan valmiiksi toteuttama slot. Tätä slottia kutsumalla kyseisen

widgetin pystyy sulkemaan. Slot metodien käyttöä ei ole rajoitettu pelkästään signals&slots –rakenteeseen. Niitä on mahdollista kutsua ohjelmakoodissa, kuten mitä tahansa tavallisia metodeja.

3.1.3 Modulaarisuus

Qt sovelluskehys on suunniteltu modulaariseksi. Täten ohjelmistot eivät julkaistessa paisu liian suuriksi, vaan kehittäjä pystyy valitsemaan mitä osia Qt:sta käyttää.

Tässä projektissa päädyttiin käyttämään kolmea eri moduulia. Tämän ansiosta ohjelmiston asennuspaketti säilyy verrattaen pienenä, jos vertailukohtana pitäisi tilannetta jossa ohjelman mukaan joutuisi pakkaamaan koko ohjelmistokehyksen.

Koko Qt sovelluskehyksen perustana toimii QtCore moduuli. Sisällyttäessä projektiin pelkästään tämän moduulin, on kehittäjän mahdollista kehittää komentorivillä ajettavia ohjelmia. QtCore sisältää Qt:n omat templaatti säiliöluokat, jotka pohjautuvat ja laajentavat C++:n Standard Template Library (STL) –kirjaston vastaavia. Näihin kuuluvat muunmuassa QList ja QVector. Lisäksi tästä moduulista löytyvät myös tiedostojen ja tiedostojärjestelmän hallintaan keskittyvät QFile ja QDir, sekä monisäikeistä ohjelmointia varten QThread apuluokka.

QtGui moduuli sisältää nimensä mukaisesti Graphical User Interfacen, eli graafisen käyttöliittymän komponentteja. Näihin valmiisiin komponentteihin voidaan lukea erilaiset valmiit ikkunatyypit ja dialogit, QMainWindow, QWidget ja QDialog. Lisäksi tämä moduuli sisältää myös komponentit kaikenlaisen grafiikan piirrolle. Tähän ryhmään lukeutuvat QGraphicsView, QPainter ja näiden apuluokat. QtGui on vastuussa myös erilaisista hiiren ja näppäimistön toimintoihin reagoivista event-handler (tapahtuman-hoitaja) metodeista, näiden ansiosta kaikki graafisen käyttöliittymän komponentit saadaan reagoimaan muun muassa pikanäppäimiin ja sisältövalikoiden näyttöpyyntöihin.

Kolmas sovelluksessa käytetty Qt:n moduuli oli QtNetwork. Tämä moduuli mahdollistaa sovelluksen kommukoinnin verkossa. Moduulin avulla voi esimerkiksi

lähettää tietopyyntöjä palvelimille QNetworkRequest –luokan avulla, ja hoitaa palvelimelta tuleva vastaus siististi toisen apuluokan QNetworkReply:n avulla. Lisäksi QtNetwork moduuli mahdollistaa QTcpSocket ja QUdpSocket olioiden luomisen. Näiden kahden luokan avulla sovelluksen ollessa yhdistettynä verkkoon, on sen mahdollista lukea ja kirjoittaa jatkuvaa tietovirtaa.

3.2 Valintakriteerit (luottamuksellinen)

3.3 Kehitystyökalut

Ohjelmistoprojekti pyrittiin tekemään käyttäen työkaluja, jotka mahdollistaisivat ohjelmiston helpon ylläpidon ja päivittämisen myös tulevaisuudessa.

3.3.1 QtCreator

Kehitysympäristöksi valittiin Qt –ohjelmistokehyksen valmistajan ylläpitämä QtCreator IDE (Integrated Development Environment), joka on osa suurempaa QtSDK (Software Development Kit) pakkausta.

Täten ohjelmiston pohjana käytetystä Qt –ohjelmistokehyksestä saatiin aina uusin versio käyttöön, kun sen ylläpitäjät sellaisen julkaisivat. Kyseisen käytännön ansiosta ohjelmiston kehityskaaren aikana on aina käytössä uusimmat ohjelmistokehyksen tarjoamat ominaisuudet.

Sovelluksen kehityksen aikana käytetyn Qt –ohjelmistokehyksen versio päivittyikin versiosta 4.7.0 versioon 4.8.0. Käytettäessä tätä ajatusmallia voidaan toki törmätä mahdollisiin yhteensopivuus ongelmiin vanhojen ominaisuuksien ja ohjelmistokehyksen uusien päivitysten kanssa. Samanaikaisesti voidaan toki ajatella, että mahdollinen tulevaisuudessa ohjelmistoa ylläpitävä taho pääsee helpommalla, koska voi vain ladata uusimman version ohjelmistokehyksestä ja jatkaa kehitystä, välttyen hankalalta tilanteelta jossa mahdollisesti muuten joutuisi metsästämään tietyn version ohjelmistokehyksestä saadakseen sovelluksen toimimaan.

3.3.2 Subversion

Yritys, jonka alaisuudessa sovelluskehitys suoritettiin, käyttää projektiensa hallintaan subversion versionhallintaohjelmistoa, joten se oli luontainen valinta tähänkin projektiin.

Yleisesti Subversion versionhallintajärjestelmää kutsutaan nimellä svn. Subversionin toiminta perustuu keskuspalvelimeen, jolla sijaitsee versioarkisto. Ottamalla tietoverkon yli yhteyden tähän versioarkistoon, voivat monet käyttäjät samanaikaisesti ottaa omille koneilleen paikalliset kopiot projektista ja tehdä siihen muutoksia. Yhden käyttäjän tehtyä haluamansa muutokset projektiin, voi hän lähettää ne takaisin versionhallintajärjestelmään, jolloin muutokset tulevat tarjolle kaikille projektissa mukana oleville tahoille.

Versionhallintaohjelmisto pitää kirjaa näistä muutoksista, ja antaa täten ohjelmoijalle mahdollisuuden palata aikaisempaan versioon näin tämän halutessa. Samanaikaisesti versionhallinta pitää kirjaa jokaisessa versiossa lisätyistä muutoksista, tämän ansiosta on helppoa katsoa missä vaiheessa ohjelmiston kehityskaarta jokin ominaisuus on toteutettu, tai että missä vaiheessa tietty ohjelmointivirhe on päässyt livahtamaan mukaan ohjelmaan.

(Julkisesta versiosta on poistettu sisällön luottamuksellisuuden vuoksi varsinainen sovelluksen suunnittelua ja toteutusta käsittelevä osio)

LÄHTEET

Qt | Luettu 6.2.2012
http://fi.wikipedia.org/wiki/Qt

Qt (framework) | Luettu 6.2.2012
http://en.wikipedia.org/wiki/Qt_(framework)

Using the Meta-Object Compiler (moc) | Luettu 6.2.2012
http://developer.qt.nokia.com/doc/qt-4.8/moc.html

Signals & Slots | Luettu 6.2.2012
http://developer.qt.nokia.com/doc/qt-4.8/signalsandslots.html

Paikkatieto | Luettu 12.2.2012
http://fi.wikipedia.org/wiki/Paikkatieto

Subversion | Luettu 12.2.2012
http://fi.wikipedia.org/wiki/Subversion