



M Ű E G Y E T E M 1 7 8 2

Budapesti Műszaki és Gazdaságtudományi Egyetem

Villamosmérnöki és Informatikai Kar

Irányítástechnika és Informatika Tanszék

Grafikus leírónyelvek közös részhalmazának gépi feldolgozása

ÖNÁLLÓ LABORATÓRIUM BESZÁMOLÓ

Készítette

Főglein Simon István

Konzulens

dr. Somogyi Péter

2023. május 19.

Tartalomjegyzék

Kivonat	3
Abstract	4
Bevezető	5
1. A Qt és GTK keretrendszerek áttekintése	6
1.1. Bemutató, rövid történet	6
1.2. Technológiai áttekintés	7
2. Fordítóprogramok	9
2.1. Fordítóprogramok bemutatása	9
2.2. Fordítóprogramok architektúrája	9
3. Saját munka bemutatása	10
Köszönetnyilvánítás	11
Irodalomjegyzék	12

HALLGATÓI NYILATKOZAT

Alulírott *Főglein Simon István*, szigorló hallgató kijelentem, hogy ezt a beszámolót meg nem engedett segítség nélkül, saját magam készítettem, csak a megadott forrásokat (szakirodalom, eszközök stb.) használtam fel. Minden olyan részt, melyet szó szerint, vagy azonos értelemben, de átfogalmazva más forrásból átvettem, egyértelműen, a forrás megadásával megjelöltem.

Hozzájárulok, hogy a jelen munkám alapadatait (szerző(k), cím, angol és magyar nyelvű tartalmi kivonat, készítés éve, konzulens(ek) neve) a BME VIK nyilvánosan hozzáférhető elektronikus formában, a munka teljes szövegét pedig az egyetem belső hálózatán keresztül (vagy autentikált felhasználók számára) közzétegye. Kijelentem, hogy a benyújtott munka és annak elektronikus verziója megegyezik. Dékáni engedéllyel titkosított diplomatervek esetén a dolgozat szövege csak 3 év eltelte után válik hozzáférhetővé.

Budapest, 2023. május 19.

Főglein Simon István
hallgató

Kivonat

Jelen dokumentum egy diplomaterv sablon, amely formai keretet ad a BME Villamosmérnöki és Informatikai Karán végző hallgatók által elkészítendő szakdolgozatnak és diplomatervnek. A sablon használata opcionális. Ez a sablon \LaTeX alapú, a *TeXLive* \TeX -implementációval és a PDF- \LaTeX fordítóval működőképes.

Abstract

This document is a \LaTeX -based skeleton for BSc/MSc theses of students at the Electrical Engineering and Informatics Faculty, Budapest University of Technology and Economics. The usage of this skeleton is optional. It has been tested with the *TeXLive* \TeX implementation, and it requires the PDF- \LaTeX compiler.

Bevezető

A dolgozat a Qt és GTK felhasználói felület keretrendszerek felületleíró nyelvei közötti átjárhatóságot mutatja be. Mindkét technológia széleskörűen elterjedt mind a FOSS, mind a kereskedelmi szoftverek körében. A FOSS projektekre jellemző forkolást, továbbfejlesztést segítené a két technológia közötti átjárhatóság. A Qt licencelése jelentősen függ a *The Qt Company*-tól, így ha ők a kizárólagos kereskedelmi licenc mellett döntenek [3], sok projekt bajba kerülhet. A GTK viszont egy tervezése óta szabad szoftverként licencelt GUI keretrendszer, mely alkalmas lehet a Qt helyettesítésére. A Qt, mint keretrendszer pedig alkalmasabb lehet egy komplex projekt megvalósítására, ugyanis az általa biztosított könyvtárak számtalan magasabb absztrakciós szintű osztályt tartalmaznak, ezzel is könnyítve a fejlesztés menetét. Egy fordítóprogram, mely elősegíti a két kezelőfelület-keretrendszer közötti átjárást nagyban segítheti egy projekt más technológiára való átalakítását.

A jelenleg elérhető megoldások nagyon kezdetlegesek, lényegében csak XML \rightarrow XML és XML \rightarrow JSON átalakítást tesznek lehetővé, nincs hatékony eljárás a két technológia közötti átjárásra.

1. fejezet

A Qt és GTK keretrendszerek áttekintése

1.1. Bemutató, rövid történet

A GTK és a Qt (ejtése mint az angol *cute* szó) széles körben elterjedt GUI eszközkészlet-keretrendszerek. Mindkettővel lehetőség van összetett felhasználói felületek készítésére, a Qt által biztosított osztályok ezen kívül lehetőséget adnak komplex alkalmazások létrehozására is.

A Qt és a GTK története is az 1990-es évekre vezethető vissza. A Qt fejlesztését 1990 nyarán kezdte meg Haavard Nord és Eirik Chambe-Eng, amikor egy ultrahangfelvételek tárolására alkalmas programot fejlesztettek [7]. Később céget alapítottak, 1994-ben megalakult a Quasar Technologies, ami később Trolltech-ként vált ismertté, manapság pedig a The Qt Company nevet viseli. A keretrendszer köré szerveződött cég jól mutatta, hogy a Qt alkotói pénzt szerettek volna keresni a könyvtárral, így a licence nem engedte a szabad terjesztést. Ez először akkor kezdett problémává válni, amikor a KDE – egy népszerű Linux asztali környezet – bebiztosította a helyét a túlnyomórészt szabad szoftverekből álló Linuxos világban.



1.1. ábra. A Qt és a GTK logója

A GTK történetének kezdete körülbelül 1996-ra tehető, ekkor kezdte meg ugyanis Peter Mattis a keretrendszer fejlesztését [5]. A cél a GIMP-hez akkor használt Motif GUI eszközkészlet lecserélése volt (a könyvtár eredeti neve, a GIMP ToolKit is innen ered), amit

végül az 1998 nyarán megjelent 1.0-s GIMP verzióval sikerült is véghezvinni.

Mindkét technológia elterjedésében jelentős szerepe volt annak, hogy a '90-es évek végén nagyobb asztali környezetek kezdték el használni mind a Qt, mind a GTK könyvtárakat. A KDE-projekt a Qt egyik legjelentősebb felhasználója és számos változtatással segítik a keretrendszer fejlődését, míg a GNOME asztali környezet fejlesztése a GTK alakulására van nagy hatással. Fontos különbség azonban a fent már említett licencelés problémája: a Qt egy kereskedelmi forgalomban lévő szoftvercsomag, mely néhány kisebb kivétellel (pl. nyílt forráskódú projektek [1], oktatási célok [2]) csak licenccím megfizetése ellenében használható. Ez sokaknak nem tetszett a szabad szoftverekben bővelkedő Unixos világban, így ez is motiválta a GTK korai fázisában a fejlesztést, ugyanis a GTK teljesen szabad licenccel rendelkezik, bárki szabadon felhasználhatja, módosíthatja és terjesztheti is.

1.2. Technológiai áttekintés

A Qt elsődleges programozási nyelve a C++, míg a GTK-é a C, bár mindkettőhöz léteznek megoldások más nyelvekkel való együttműködés biztosítására is, mint például Python és C++.

A felhasználói felületek leírásához alapvetően mindkét technológia XML-alapú megoldást használ, bár a Qt esetében lehetőség van a JSON-alapú QML használatára is. A QML (Qt Modeling Language) egy deklaratív felhasználófelület-leíró nyelv, melyet a Nokia fejlesztett a Qt projekthez 2009 környékén [6]. Előnye az XML-alapú megoldáshoz képest, hogy könnyebben áttekinthető, valamint lehetőséget biztosít JavaScript használatára is, így például a KDE számos alkalmazását folyamatosan portolják át a hagyományos C++ és XML technológia helyett a QML-esre (ugyanakkor a QML részei is elérhetőek C++ kódból, például lehetőség van eseménykezelők regisztrálására is).

QML	Qt XML	GTK XML
<pre> 1 QWidget { 2 name: "centralWidget" 3 QPushButton { 4 name: "pushButton" 5 geometry: { 6 x: 10 7 y: 20 8 width: 150 9 height: 50 10 } 11 text: "Hello World!" 12 } 13 }</pre>	<pre> 1 <widget class="QWidget" 2 name="centralWidget"> 3 <widget class="QPushButton" 4 name="pushButton"> 5 <property name="geometry"> 6 <rect> 7 <x>10</x> 8 <y>20</y> 9 <width>150</width> 10 <height>50</height> 11 </rect> 12 </property> 13 <property name="text"> 14 <string>Hello World!</string> 15 </property> 16 </widget> 17 </widget></pre>	<pre> 1 <object class="GtkFrame"> 2 <child> 3 <object class="GtkButton" 4 id="pushButton"> 5 <property name="label"> 6 translatable="yes"> 7 Hello World! 8 </property> 9 <property name="name"> 10 pushButton 11 </property> 12 </object> 13 </child> 14 </object></pre>

2. fejezet

Fordítóprogramok

2.1. Fordítóprogramok bemutatása

A fordítóprogramok olyan számítógépes szoftverek, amelyek egy adott programozási nyelven írt programot képesek egy másik programozási nyelvre, vagy számítógépek által értelmezhető, futtatható gépi kódra átalakítani.

2.2. Fordítóprogramok architektúrája

3. fejezet

Saját munka bemutatása

A fordítóprogramok nagyon komplex szoftverek. A legismertebb C/C++ fordítóprogram talán a GCC (GNU Compiler Collection), amely óriási kódbázissal rendelkezik, 2019-ben körülbelül 15 millió sort tartalmaztak a forrásállományai [4]. Természetesen egy GCC szintű fordítóprogram számos olyan funkcióval rendelkezik, mely a több, mint 36 évnyi fejlesztésből és a projekt léptékéből adódik, ilyen például a többféle programozási nyelv támogatása, valamint a fejlett optimalizációs megoldások. Ezen funkciók közül jó néhányat egy felhasználóifelület-leíró nyelveket támogató fordítóprogramnak nem szükséges biztosítania, és mivel az Önálló laboratórium tárgy választott témájának keretében a fordítóprogramok felépítésének és készítésének, valamint a programozási nyelvek konstrukcióinak mélyebb megismerése állt a középpontban, így elsősorban ezekre fektettem nagy hangsúly az irodalomkutatás során, továbbá ilyen területekre fogok összpontosítani a fordítóprogramom implementálása alatt is.

Köszönetnyilvánítás

Ez nem kötelező, akár törölhető is. Ha a szerző szükségét érzi, itt lehet köszönetet nyilvánítani azoknak, akik hozzájárultak munkájukkal ahhoz, hogy a hallgató a szakdolgozatban vagy diplomamunkában leírt feladatokat sikeresen elvégezze. A konzulensnek való köszönetnyilvánítás sem kötelező, a konzulensnek hivatalosan is dolga, hogy a hallgatót konzultálja.

Irodalomjegyzék

- [1] Qt Group. *Qt for Open Source Development*, May 2023. URL: <https://www.qt.io/download-open-source>.
- [2] Qt Group. *Qt Educational license for students and teachers*, May 2023. URL: <https://www.qt.io/qt-educational-license>.
- [3] Olaf Schmidt-Wischhöfer. *Qt, Open Source and corona*. KDE, April 2020. URL: <https://mail.kde.org/pipermail/kde-community/2020q2/006098.html>.
- [4] Wikipedia. *GNU Compiler Collection*, May 2023. URL: https://en.wikipedia.org/wiki/GNU_Compiler_Collection.
- [5] Wikipedia. *GTK*, May 2023. URL: <https://en.wikipedia.org/wiki/GTK>.
- [6] Wikipedia. *QML*, May 2023. URL: <https://en.wikipedia.org/wiki/QML>.
- [7] Wikipedia. *Qt (Software)*, May 2023. URL: [https://en.wikipedia.org/wiki/Qt_\(software\)](https://en.wikipedia.org/wiki/Qt_(software)).