

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
FAKULTA STAVEBNÍ



DIPLOMOVÁ PRÁCE

Workflow builder pro Quantum GIS

Vypracoval: Zdeněk Růžička

Vedoucí práce: Ing. Martin Landa

Rok: Praha, 2012

Prohlášení

Prohlašuji, že jsem svou diplomovou práci na téma **Workflow Builder** vypracoval samostatně s pomocí svého vedoucího práce a za použití literatury a zdrojů uvedených v příloženém seznamu na konci práce.

V Praze dne

podpis

Poděkování

Především děkuji vedoucímu mé diplomové práce Ing. Martinu Landovi, Ph.D. za odborné vedení, rychlé reakce na mé dotazy a ochotu hledat na ně odpovědi. Dále bych chtěl poděkovat Camilo Polimeris za napsání QGIS Processing Frameworku jehož je tato práce součástí. V neposlední řadě bych chtěl poděkovat rodině a kamarádům za důvěru a podporu během studií.

Abstrakt

Diplomová práce si vytyčila za cíl vytvořit v prostředí Quantum GIS (QGIS) nástroj, který by umožňoval uživateli grafické propojování modulů z frameworku **QGIS Processing Framework**. V úvodních dvou kapitolách je představeno prostředí QGIS a popsána práce s QGIS Processing Frameworkem jak z pozice uživatele GISu, tak z pozice vývojáře nových modulů. Dále je zmíněn SAGA Plugin, který byl napsán v rámci onoho frameworku.

V dalších částech diplomové práce představím z rychlíku knihovnu Qt, resp. její verzi PyQt pro jazyk Python, ve kterém byl celý Workflow Builder napsán.

V poslední kapitole diplomové práce je představena samotná aplikace umožňující grafické propojování modulů z QGIS Processing Framework Manageru do grafu, jejich postupné spuštění a případné uložení jako nový modul.

Klíčová slova

Quantum QGIS, workflow, open source, GIS, PyQt, SAGA, QGIS Processing Framework

Abstract

Key words

Quantum QGIS, workflow, open source, GIS, PyQt, SAGA, QGIS Processing Framework

Obsah

Úvod	1
1 Teorie	3
1.1 Quantum GIS	3
1.1.1 Správa pluginů	4
1.1.2 Psaní vlastního pluginu	5
1.1.3 Python plugin	7
1.1.4 C++ plugin	7
1.2 QGIS Processing Framework	8
1.2.1 Psaní pluginu	8
1.3 Python	9
1.4 Qt, PyQt	10
1.4.1 Model/View Programming	11
1.4.2 Graphics View Interface	11
1.4.3 VisTrails, Orange	11
1.5 .xml.dom.monidom	12
2 Workflow Builder	13
2.1 Tvorba workflow	13
2.2 Uložení workflow	13
2.2.1 Popsání výstupního xml souboru	13
2.3 Načtení workflow do PF Manageru	14

3	SEXTANTE	15
3.1	Srovnání QGIS Processing Framework v SEXTANTE	15
3.2	Srovnání Workflow Builder v SEXTANTE Modeler	15
	Rejstřík	I
	Literatura	III

Úvod

V dnešní době se můžeme setkat s geoinformačními (GIS) technologiemi na každém kroku. V různých oblastech krajinného inženýrství, při plánování výstavby silnic, v územním plánování, při řešení krizových situací či plánování záchranných akcí. Uživatel si může vybrat z nepřeberného množství již existujících GIS nástrojů, řešení. A je pěkné, že svobodná řešení, nejen v oblasti geoinformačních technologií, drží krok s těmi proprietárními. Uživatel tedy nemusí sahát hluboko do kapsy. Co se týče nástrojů pro prohlížení, zpracování a analýzu geodat, můžeme jmenovat například GRASS GIS, gvSIG, Quantum GIS či SAGA GIS. Tato práce si ale nekladla za cíl srovnat GIS nástroje, ale implementací nástroje do programu Quantum GIS, který by uživateli umožňoval vytvářet vlastní funkce spojováním již existujících funkcí.

Můžeme se také setkat s pojmy jako `model builder` či `chaining`, v té práci budu používat pojem `workflow builder`. Tento název byl převzat z projektu VisTrails, který byl inspirací pro grafiku. Takzvané `workflow buildery` dávají uživateli možnost vytvářet si vlastní moduly za pomoci spojování výstupů a vstupů modulů již existujících. Uživatel tak nemusí spouštět každý modul zvlášť a starat se o výstupy, nová data, která se vytvoří jen dočasně a která uživatel v konečném výsledku nepotřebuje. Dále je pro uživatele je mnohem pohodlnější, pakliže může najít všechny funkce na jednom místě (tzv. `toolbox`), nežli při hledání procházet všechny možné `plugins`.

V době psaní této diplomové práce existoval projekt QGIS Processing Framework studenta Camilo Polymeris z univerzity Universidad de Concepción. QGIS Processing Framework si kladl za cíl být frameworkem, který by sdružoval moduly z `pluginů` pro QGIS na jednom místě (tzv. `toolbox`). Odtud by byly jednotlivé moduly volány, pomocí `workflow builderu` spojovány, ukládány atp. V rámci tohoto projektu začala

vznikat podpora pro použití modulů z jiného GIS nástroje - System for Automated Geoscientific Analysis (SAGA GIS). V době psaní existovala podpora pro 170 modulů, ne všechny ale byly testovány a fungovaly správně. Ale i přesto se mohlo začít s prací na workflow builderu.

Aktuální verzi workflow builderu můžete najít zde:

<https://github.com/CzendaZdenda/qgis>

Kapitola 1

Teorie

V této kapitole představím jeden ze svobodných systémů pro práci s geografickými daty - Quantum GIS a možnost rozšiřování funkcionality pomocí zásuvných modulů, tzv. pluginů. V druhé části popíši projekt QGIS Processing Framework, co bylo jeho cílem a také možnosti jeho rozšiřování. Dále se stručně charakterizuji jazyka Python, ve kterém vznikal Workflow Builder. V další části se budu věnovat knihovně Qt, respektive její verze pro Python - PyQt, kde popíši nástroje, které jsem využil při psaní Workflow Builderu. Mezi tyto nástroje patří Model/View Programming a Graphics View Framework. V poslední části představím modul xml.dom jazyka Python pro práci s objekty ve formátu XML formátem.

1.1 Quantum GIS



Quantum GIS je nejen prohlížečka geografických dat dostupná pro MS Windows, GNU/Linux, Unix, BSD a Mac OS X. Quantum GIS podporuje díky knihovně OGR většinu vektorových formátů dat jako například ESRI Shapefile, GRASS, MapInfo či GML a díky knihovně GDAL mnoho rastrových formátů jako TIFF, ArcInfo, GRASS

raster, ERDAS a další. Přes Quantum GIS můžeme také přistupovat k datům uložených v geodatabázích PostGIS a SpatiaLite či k datům dostupných přes WMS a WFS služby. Quantum GIS je šířen pod licencí GNU Public Licence.¹

Program je napsán v jazyce C++. Poslední stabilní verze nese označení 1.7.4. Quantum GIS je lehce rozšiřitelný program pomocí pluginů, které mohou být psány na jazyce Python nebo C++. Quantum GIS má poměrně dobře zdokumentované API a nutno také podotknout, že komunita kolem Quantum GISu je aktivní a podpora skrze mailinglisty je na vysoké úrovni.

Systém začal vyvíjet v roce 2002 Gary Sherman. Mělo jít o nenáročnou prohlížečku geodat pro Linux s širokou podporou datových formátů. Dlouhou dobu byl Quantum GIS brán převážně jako grafická nadstavba pro jiný desktopový GIS - GRASS GIS. Přes GRASS Plugin QGIS zpřístupňuje řadu modulů GRASS GIS. V současnosti se na vývoji nejvíce podílí skupina vývojářů kolem organizace² Faunalia.

Funkcionalitu Quantum GIS rozšiřuje množství pluginů. Jako základní pluginy bych označil³ **fTools**, který umožňuje pokročilé prostorové analýzy nad vektorovými daty,⁴ **GdalTools** pro práci s rastrovými daty a již zmíněný⁵ **GRASS Plugin** plugin, který zpřístupňuje funkce GRASSu uživatelům Quantum GIS.

1.1.1 Správa pluginů

Jak už bylo zmíněno Quantum GIS umožňuje uživatelům rozšiřovat funkce programu dle jejich potřeb v podobě zásuvných modulů. Díky dobře zdokumentovanému API může uživatel pohodlně psát pluginy v jazyce C++ nebo Python. Pluginy píše jak vývojáři Quantum GISu, tak i obyčejní uživatelé. Pluginy si můžete stáhnout z oficiálních či neoficiálních repositářů. Pro instalování pluginů napsaných v jazyce Python a správu repositářů slouží nástroj **QGIS Python Plugin Installer**, dostupný přes *Plugins → Fetch Python Plugins...*

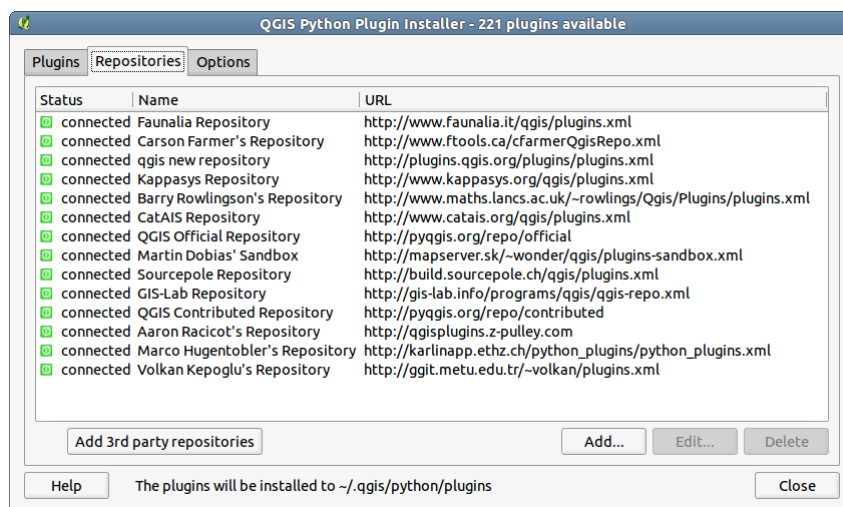
¹<http://qgis.org/about-qgis/features.html>

²<http://www.faunalia.co.uk/quantumgis>

³<http://www.ftools.ca/>

⁴<http://www.faunalia.co.uk/gdaltools>

⁵http://grass.osgeo.org/wiki/GRASS_and_QGIS



Obrázek 1.1: QGIS Python Plugin Installer - správa repozitářů

Jak je vidět z [Obr. 1.1], takto nainstalované pluginy se stáhnou do adresáře:

- `$HOME/.qgis/python/plugins` - v případě OS GNU/Linux
- `C:\Documents and Settings\USER\.qgis/python/plugins` - v případě OS Windows bývá cesta podobná této

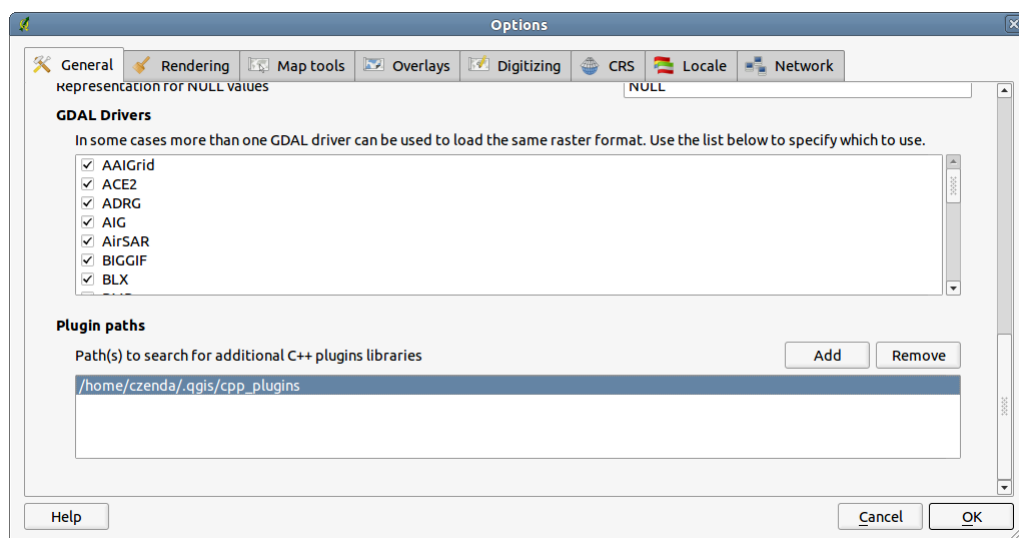
Pakliže uživatel napíše plugin v jazyce Python, doporučuji ho uložit do tohoto adresáře. Je také sice možnost uložit plugin do adresáře `$QGIS_INSTALL_DIR\share\qgis/python/plugins` ale při případné opětovné kompilaci by byly změny ztraceny.

Pluginy psané v C++ se po přeložení ukládají standardně v `$QGIS_INSTALL_DIR\lib\qgis/plugins` případně uživatel může přidat nová úložiště pomocí *Settings* → *Options* a v záložce *Generals* zadat cestu [Obr. 1.2].

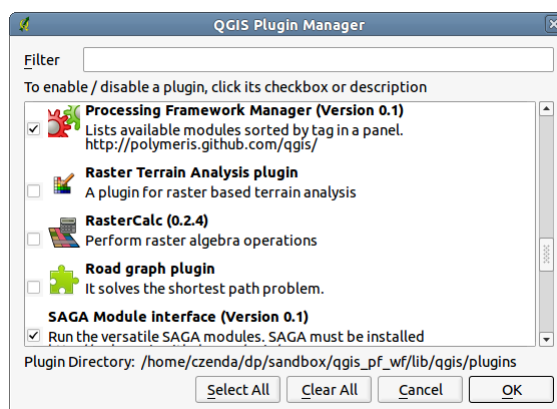
Všechny nainstalované pluginy, ať psané v jazyce C++ či Python, může uživatel spravovat přes **QGIS Plugin Manager** - *Plugins* → *Manage Plugins...* [Obrázek 1.3.

1.1.2 Psaní vlastního pluginu

Pluginy mohou být psány na jazyce C++ a Python. Již z charakteristiky daných jazyků vyplývá, že pro jednoduché, nenáročné či na začátku vývoje pluginy, se bude hodit spíše jazyk Python, který se nemusí kompilovat a píše se v něm rychleji než v



Obrázek 1.2: *Settings*→*Options*→*Generals* - přidání nové cesty k pluginům psaných na jazyce C++



Obrázek 1.3: *Plugins* → *Manage Plugins...* - správa pluginů

jazyce C++. Pro rozsáhlejší projekty je lepší sáhnout po jazyce C++. Obecně jsou programy psané v kompilovaných jazycích mnohem rychlejší než programy psané v jazycích interpretovaných.

1.1.3 Python plugin

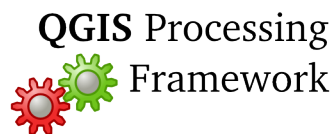
Při psaní pluginu v jazyce Python využíváme nástroje PyQGIS. Kromě dokumentace k Quantum GIS API také doporučuji [3]. Dále můžeme využít nástroj Plugin Builder, což je v podstatě také plugin, který vygeneruje základní soubory, kód, který potom začneme upravovat podle tak, aby náš plugin dělal to, co chceme.

```
1 def name():
2     return "My testing plugin"
3
4 def description():
5     return "Popis pluginu."
6
7 def version():
8     return "Version 0.1"
9
10 def qgisMinimumVersion():
11     return "1.0"
12
13 def authorName():
14     return "Developer"
15
16 def classFactory(iface):
17     # load TestPlugin class from file testplugin.py
18     from plugin import TestPlugin
19     return TestPlugin(iface)
```

Ukázka kódu 1.1: __init__.py - inicializační soubor

1.1.4 C++ plugin

1.2 QGIS Processing Framework



QGIS Processing Framework vznikl v rámci projektu ⁶GSoC 2011 . Student Camilo Polymeris z univerzity Universidad de Concepción si kladl za cíl napsat framework, který ...

1.2.1 Psaní pluginu

⁶Google Summer of Code. Projekt společnosti Google na podporu studentu. více na <http://code.google.com/soc/>

1.3 Python

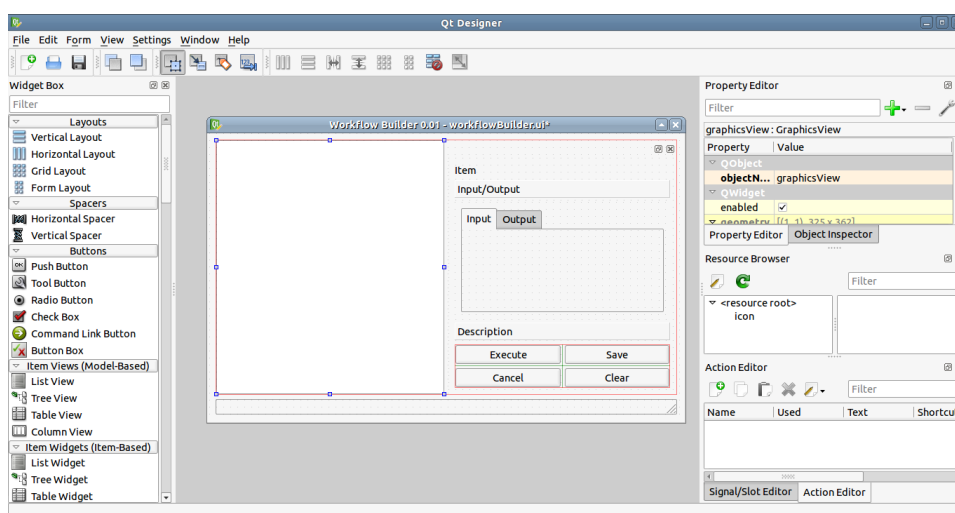
Python je interpretovaný, procedurální a objektově orientovaný jazyk, ve kterém se rychle programuje. Existuje pythoní verze Qt - PyQt.

1.4 Qt, PyQt



V současné době se vývojem Qt zabývá firma Nokia, která Qt koupila v roce 2008 od společnosti Trolltech. Společnost Trolltech započala s vývojem Qt v roce 1999. Qt je poměrně mocný soubor nástrojů pro psaní grafických aplikací v jazyce C++. Není to pouze knihovna pro psaní GUI. Qt nabízí také řadu programů, které usnadňují vývojáři práci. Například velmi kvalitní IDE v podobě Qt Creator či Qt Designer pro pohodlnou tvorbu grafického rozhraní pouhým přetahováním widgetů myší. Qt Designer umožňuje pohodlně rozvrhnout a umístit jednotlivé widgety, seskupovat je do layoutů či nastavovat parametry.

Existuje mimo jiné verze pro Python - PyQt. PyQt je vyvíjena firmou Riverbank Computing. Z rodiny Qt, resp. PyQt, se v této práci využila samotná knihovna pro psaní kódu, obzvláště pak její Graphics View Interface, a program QtDesigner.



Obrázek 1.4: Qt Designer - nástroj pro tvorbu grafického rozhraní

Soubor vytvořený v Qt Designer (s příponou **.ui**) lze jednoduše přeložit programem **pyuic4** do pythoního kódu, který lze poté dále použít.

1

```
pyuic4 soubor_vytvoreny_v_Qt_Designer.ui -o soubor_v_Pythonu.py
```

Ukázka kódu 1.2: pyuic4 - přeložení .ui souboru do pythoního kódu

1.4.1 Model/View Programming

1.4.2 Graphics View Interface

1.4.3 VisTrails, Orange

1.5 .xml.dom.monidom

Nově vzniklé workflow se ukládají ve formátu XML. Výhoda tohoto formátu je, že se s ním snadno pracuje a jde v podstatě pouze o textový dokument.

Python nabízí několik možností pro zpracování XML dokumentů.

XML nabízí jednoduché uložení hierarchicky strukturovaných dat. O prvcích XML dokumentu hovoříme jako elementech. Elementy jsou ohraničeny počátečními a koncovými znaky, tzv. tagy. XML dokument obsahuje vždy právě jeden kořenový element. Ten se může skládat z dalších a dalších elementů. Elementy mohou obsahovat atributy (dvojice jméno="hodnota"). Atributy se nesmějí v opakovat. Elementy kromě toho mohou obsahovat text. Text se uvádí mezi počáteční a koncový znak.

Příklad XML dokumentu:

```
1 <Graph name="Addition two rasters">
2   Description ...
3   <SubGraph id="17">
4     <Module id="769" name="Input raster by path">
5       You can register raster layer to QGIS by giving the path.
6       <tag> workflow builder </tag>
7     </Module>
8     <Module id="998" name="Operations with two rasters">
9       Pixel by pixel operations with two rasters.
10    </Module>
11  </SubGraph>
12 </Graph>
```

Ukázka kódu 1.3: Příklad XML dokumentu

Kapitola 2

Workflow Builder

2.1 Tvorba workflow

2.2 Uložení workflow

2.2.1 Popsání výstupního xml souboru

XML nabízí jednoduché uložení hierarchicky strukturovaný dat. O prvcích XML dokumentu hovoříme jako elementech. Elementy jsou ohraničeny počátečními a koncovými značkami, tzv. tagy. XML dokument obsahuje vždy právě jeden kořenový element. Ten se může skládat z dalších a dalších elementů. V našem případě je kořenový element Graph. Ten se skládá z minimálně jednoho podgrafu (SubGraph), a ten poté minimálně z jednoho modulu (Module). Podgraf dále může obsahovat spojení mezi moduly (Connection). Modul kromě toho obsahuje elementy parametr (Port) a tag (tag) a popis. Graf také obsahuje tagy a popis.

atribut	příklad
name	Addition two rasters
tags	['raster', 'hydrology']

Tabulka 2.1: atributy elementu Graph

2.3 Načtení workflow do PF Manageru

Kapitola 3

SEXTANTE

3.1 Srovnání QGIS Processing Framework v SEXTANTE

3.2 Srovnání Workflow Builder v SEXTANTE Modeler

Závěr

Dále musím zmínit existenci druhého frameworku, který se objevil v konci psaní této práce. SEXTANT pro Quantum GIS. Daný framework má v podstatě podobné cíle a v současné době se zdá být on tou pravou cestou pro qgis, ikdyž ne všechny moduly jsou momentálně plně funkční.

Ukázky kódu

1.1	__init__.py - inicializační soubor	7
1.2	pyuic4 - přeložení .ui souboru do pythoního kódu	11
1.3	Příklad XML dokumentu	12

Rejstřík

Faunalia, 4

fTools, 4

GDAL, 3

GDAL, GdalTools, 4

geodata, 1

GIS, 1

GRASS Plugin, 4

OGR, 3

PyQGIS, 7

Python, 9

QGIS Processing Framework, 1

QGIS,Quantum GIS, 3

SAGA GIS, 2

SEXTANT, 16

VisTrails, 1

XML, 12

Bibliography

- [1] *Official homepage of Quantum GIS*. URL: <http://www.qgis.org>.
- [2] Mark Pilgrim. *Ponořme se do Python(u) 3. Dive Into Python 3*. CZ.NIC, z. s. p. o., 2010. URL: <http://qgis.org/api/>.
- [3] *PyQGIS Developer Cookbook*. 2012. URL: <http://www.qgis.org/pyqgis-cookbook/index.html>.
- [4] *PyQt Class Reference*. URL: <http://www.riverbankcomputing.co.uk/static/Docs/PyQt4/html/classes.html>.
- [5] *Python v2.7.3 documentation*. URL: <http://docs.python.org/>.