

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
FAKULTA STAVEBNÍ

DIPLOMOVÁ PRÁCE

2016

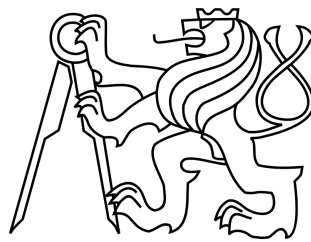
Bc. Štěpán BAMBULA

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

FAKULTA STAVEBNÍ

STUDIJNÍ PROGRAM: GEODÉZIE A KARTOGRAFIE

STUDIJNÍ OBOR: GEOMATIKA



DIPLOMOVÁ PRÁCE

ROZŠÍŘENÍ NÁSTROJE PRO PRÁCI
S KATASTRÁLNÍMI DATY V PROGRAMU QGIS

QGIS VFK PLUGIN IMPROVEMENTS

Vedoucí práce: Ing. Martin LANDA, Ph.D.

Katedra geomatiky

2016

Bc. Štěpán BAMBULA



ZADÁNÍ DIPLOMOVÉ PRÁCE

I. OSOBNÍ A STUDIJNÍ ÚDAJE

Příjmení: Bambula

Jméno: Štěpán

Osobní číslo: 381444

Zadávací katedra: Katedra geomatiky

Studijní program: Geodézie a kartografie

Studijní obor: Geomatika

II. ÚDAJE K DIPLOMOVÉ PRÁCI

Název diplomové práce: Rozšíření nástroje pro práci s katastrálními daty v programu QGIS

Název diplomové práce anglicky: QGIS VFK Plugin Improvements

Pokyny pro vypracování:

Cílem diplomové práce je rozšířit projekt laboratoře OSGeoREL ČVUT v Praze zaměřený na práci s katastrálními daty poskytovanými ve výměnném formátu VFK v prostředí open source nástroje QGIS. Práce navazuje na již existující nástroj implementovaný jako tzv. zásuvný modul a rozšiřuje ho o novou funkcionalitu a to především zpracování a vizualizaci datových vět změnových souborů VFK. Druhotným cílem je usnadnění distribuce zásuvného modulu v prostředí QGIS s důrazem na jeho přenositelnost.

Seznam doporučené literatury:

Kurt Menke, G.: Mastering QGIS, Packt Publishing, 2015, ISBN: 9781784390068

Pilgrim, M.: Dive Into Python, Createspace Independent Pub 2009, ISBN: 9781441413024

Summerfield, M.: Rapid GUI Programming With Python and Qt, Prentice Hall, 2015, ISBN: 9780134393339

Jméno vedoucího diplomové práce: Ing. Martin Landa, Ph.D.

Datum zadání diplomové práce: 22.2.2016

Termín odevzdání diplomové práce: 22.5.2016

Podpis vedoucího práce

Podpis vedoucího katedry

III. PŘEVZETÍ ZADÁNÍ

Beru na vědomí, že jsem povinen vypracovat diplomovou práci samostatně, bez cizí pomoci, s výjimkou poskytnutých konzultací. Seznam použité literatury, jiných pramenů a jmen konzultantů je nutné uvést v diplomové práci a při citování postupovat v souladu s metodickou příručkou ČVUT „Jak psát vysokoškolské závěrečné práce“ a metodickým pokynem ČVUT „O dodržování etických principů při přípravě vysokoškolských závěrečných prací“.

Datum převzetí zadání

Podpis studenta(ky)

ABSTRAKT

Diplomová práce rozšiřuje stávající projekt laboratoře OSGeoREL ČVUT v Praze, který je zaměřen na práci s katastrálními daty poskytovanými ve výměnném formátu VFK v prostředí open source nástroje QGIS. Práce tento projekt, který je v QGIS implementován jako tzv. zásuvný modul, doplňuje o novou funkcionalitu, kterou je zpracování a vizualizace datových změnových vět VFK. S tímto souvisí i nezbytné změny na straně ovladače VFK v knihovně GDAL, který zásuvný modul pro čtení dat používá. Druhotným cílem diplomové práce je usnadnění distribuce zásuvného modulu, což souvisí s přepisem stávajícího projektu do jazyka Python.

KLÍČOVÁ SLOVA

VFK, NVF, QGIS, ISKN, Python, PyQt, GDAL, zásuvný modul

ABSTRACT

Diploma thesis extends the existing project of OSGeoREL at the Czech Technical University in Prague which is focused on working with the cadastral data provided in exchange format VFK in an environment of open source software QGIS. The project which is implemented in QGIS as a so-called plugin is in this thesis extended with new functionality – processing and visualisation of amendment records of VFK. It is related to the necessary changes of the VFK driver in GDAL library which plugin uses for reading data. Another goal of this study is to facilitate the distribution of plugin which is related to a transcript of an existing project to Python programming language.

KEYWORDS

VFK, NVF, QGIS, ISKN, Python, PyQt, GDAL, plugin

PROHLÁŠENÍ

Prohlašuji, že jsem diplomovou práci na téma „Rozšíření nástroje pro práci s katastrálními daty v programu QGIS“ vypracoval samostatně. Všechny podklady, ze kterých jsem čerpal, jsou uvedeny v seznamu použité literatury.

V Praze dne

.....
Štěpán Bambula

PODĚKOVÁNÍ

Na tomto místě bych chtěl poděkovat vedoucímu diplomové práce Ing. Martinu Landovi, PhD. za pomoc, připomínky a poskytnuté rady při zpracování této práce. Dále bych chtěl poděkovat Jiřímu Chroustovi za rady ohledně zpracování změnových vět. V neposlední řadě děkuji rodině za podporu během celého studia.

Seznam použitých zkratk

VFK Výměnný formát katastru nemovitostí

NVF Nový výměnný formát

ČUZK Český úřad zeměměřický a katastrální

GDAL Geospatial Data Abstraction Library

GIS Geografický informační systém

OSGeo Open Source Geospatial Foundation

ISKN Informační systém katastru nemovitostí

SGI Soubor geodetických informací

SPI Soubor popisných informací

API Application Programming Interface

Obsah

Úvod	1
1 Rešerše nástrojů pro práci s VFK	3
1.1 ISKN Studio pro ArcGIS	3
1.1.1 ISKN View pro ArcGIS	4
1.2 Import VFK, VKM	5
1.3 Spirit VFK	6
1.4 VFK2DWG	7
1.5 VFK2DB	7
1.6 Topol VFK Import	8
1.7 GDAL – VFK Driver	9
2 Použité technologie	11
2.1 QGIS	11
2.2 GDAL	12
2.3 Python	12
2.4 PyQt	13
3 Informační systém katastru nemovitostí	14
3.1 Historie a vývoj	14
3.2 Hlavní charakteristiky ISKN	15
3.2.1 Optimalizace uložení dat	15
3.2.2 Optimalizace procesů při správě KN	16
3.2.3 Bezpečnost	16
3.3 Poskytování dat	16

3.3.1	Poskytování dat dálkovým přístupem	17
3.3.2	Poskytování dat ve výměnném formátu ISKN	17
4	Výměnný formát ISKN	18
4.1	Vývoj formátu	18
4.1.1	Výměnný formát KN před ISKN	18
4.1.2	Výměnný formát VF ISKN	19
4.2	Struktura výměnného formátu ISKN	20
4.2.1	Hlavička &H	21
4.2.2	Datové bloky	23
4.2.3	Koncový znak &K	29
4.3	Změnové věty v NVF	29
4.3.1	Obsah změnového exportu – Typy tabulek	30
5	VFK Plugin pro QGIS	32
5.1	Funkcionalita	33
5.2	Použití pluginu	35
5.2.1	Instalace a spuštění	35
5.2.2	Rozložení	35
5.2.3	Načítání dat	36
5.2.4	Vyhledávání	36
6	Rozšíření stávající funkcionality	37
6.1	Uspřádání distribuce pluginu	37
6.1.1	Přepis do jazyka Python	38
6.1.2	Instalace zásuvného modulu v QGIS	39
6.2	Úprava GDAL VFK driveru	40
6.2.1	Načítání dat z více souborů	41
6.2.2	Čtení z databáze SQLite	42
6.2.3	Vytváření geometrie prvků	42
6.3	Nově přidané funkcionality	43
6.3.1	Čtení z databáze SQLite	43
6.3.2	Načítání více souborů	43

6.3.3	Načítání VFK souborů z adresáře	43
6.3.4	Automatické pojmenování databáze	44
6.3.5	Přepsání existující databáze	44
6.3.6	Změna vzhledu zásuvného modulu	44
6.4	Zpracování změnových vět	45
6.4.1	Metodika aplikace změn na stavová data	45
6.4.2	Implementace na straně zásuvného modulu	49
Závěr		50
A	Popis tříd zásuvného modulu	I
B	Ukázky kódu	V
B.1	Konstruktor třídy BudovySearchForm	V
B.2	Metoda buildHtml	V
B.3	Načtení souborů v separátním vlákně	VI
B.4	Vyhledání parcel – ukázka tvorby URL	VI
C	Uživatelská příručka	VIII
C.1	Instalace a spuštění zásuvného modulu	VIII
C.2	Rozložení zásuvného modulu	X
C.3	Načítání dat	X
C.3.1	Načítání ze souborů	X
C.3.2	Načítání z adresáře	XI
C.3.3	Možnosti načítání	XI
C.4	Vyhledávání	XI
C.5	Export vyhledaných informací	XIII
C.6	Zpracování změn	XIII
D	Obsah příloženého CD	XV

Úvod

Katastr nemovitostí je, co se týče objemu uchovávaných dat, jedním z nejrozsáhlejších informačních systémů ve státní správě. Proto i jeho data jsou velice zajímavým zdrojem informací, ať už pro uživatele ze státní sféry, či pro společnosti zabývající se například územním plánováním. Vzhledem k tomu, že data obsahují nejen popisnou, ale i prostorovou složku, tak na nich může být prováděno široké spektrum prostorových analýz. Data jsou k dispozici ve strukturovaném výměnném formátu (VFK).

Pro jejich zpracování je k dispozici několik nástrojů. Většinou se však jedná o komerční nástroje, které jsou šířeny za úplaty, nebo nástroje, které jsou dostupné jako modul pro komerční GIS software typu ArcGIS, AutoCAD Map 3D, Kokeš a další.

Alternativou k placeným programům pro práci s geografickými daty je, v posledních letech čím dál úspěšnější, projekt QGIS, který je šířen zdarma jako open-source. Projekt je úspěšný i díky mnoha dobrovolníkům, kteří se na jeho vývoji podílí. Nemalý podíl na oblíbenosti tohoto open-source programu má i knihovna GDAL, díky které je QGIS schopen číst či zapisovat do širokého spektra vektorových, ale i rastrových formátů.

Pro Českou republiku je jednou z nejdůležitějších součástí knihovny GDAL ovladač VFK, který podporuje čtení právě ze souborů vyexportovaných z Informačního systému katastru nemovitostí v tzv. novém výměnném formátu (VFK). Spolu s vývojem knihovny GDAL se vyvíjí i samotný ovladač. Od své první verze v roce 2011 byl obohacen o spoustu nových funkcionalit. Jednou z nich je například podpora čtení z databáze SQLite, vznikající při čtení dat VFK knihovnou GDAL. Tato funkcionality byla přidána i díky této diplomové práci.

Pro usnadnění práce s daty ve výměnném formátu katastru nemovitostí v prostředí QGIS, vznikl na Fakultě stavební ČVUT v Praze roku 2012 zásuvný modul, jehož autory jsou absolventi oboru Geoinformatika – Anna Kratochvílová a Václav Petráš. Tento modul byl napsán v jazyce C++ a obsahoval široké spektrum nástrojů, díky kterým bylo možné soubory VFK načíst, zobrazit v mapovém okně QGIS a provádět nad nimi vyhledávací dotazy, včetně možnosti exportu výsledků do některého z podporovaných formátů.

Hlavní nevýhodou výše zmíněného nástroje byl paradoxně použitý jazyk C++, díky němuž byla instalace a zprovoznění zásuvného modulu v QGIS ne zcela intuitivní a i pro zkušeného uživatele mnohdy představovala nepřekonatelný problém. Na základě tohoto podnětu vzniklo jedno z podtémat diplomové práce, a sice přepis stávajícího zásuvného modulu do jazyka Python. Tento jazyk byl zvolen, jelikož program QGIS podporuje zásuvné moduly napsané právě v těchto dvou jazycích. Moduly napsané v jazyce Python je možné instalovat standardní cestou, čili ze správce zásuvných modulů, což velice usnadňuje jejich dostupnost. Jazyk C++ je vhodný především u modulů, které jsou součástí jádra QGIS (např. QGIS GRASS Plugin).

V teoretické části diplomové práce se zaměřím na nezbytnou rešerši nástrojů, které mohou být pro zpracování katastrálních dat použity. Uvedu zde zástupce jak z komerční sféry, tak nástroje, které jsou dostupné bezúplatně – především tedy výše zmíněný ovladač VFK knihovny GDAL. V další kapitole teoretické části se budu věnovat popisu použitých technologií, bez kterých by diplomová práce nemohla být vypracována.

V posledních dvou částech teoretického úvodu se budu věnovat Informačnímu systému katastru nemovitostí, jež je zdrojem dat ve formátu VFK, a popisu výměnného formátu. Při tomto popisu budu čerpat především z oficiální dokumentace, která je k němu dostupná, s tím, že zde uvedu pouze nezbytné informace k pochopení struktury, do které jsou data exportována. Samozřejmě zde popíši i uchování a strukturu změnových vět, jejichž zpracování je nosným tématem diplomové práce.

V praktické části nejprve přiblížím stávající zásuvný modul napsaný v jazyce C++, kde se zaměřím na jeho instalaci v prostředí QGIS a především na jeho funkcionalitu. Po této kapitole plynule naváží na rozšíření stávající funkcionality. Zde budou nejprve uvedeny kroky související s usnadněním distribuce modulu. Kapitola bude pokračovat změnami, které musely být provedeny na straně ovladače VFK knihovny GDAL tak, aby byl zásuvný modul schopen číst a zpracovávat změnové věty výměnného formátu. O implementaci zpracování změnových vět pojednává další část kapitoly.

Výstupem práce tedy bude přepsaný zásuvný modul do programovacího jazyka Python, vhodně doplněný o nové funkcionality. Jednou z nejdůležitějších funkcionalit bude zpracování změnových vět výměnného formátu a jejich aplikace na stavová data uložena v databázi SQLite.

Kapitola 1

Rešerše nástrojů pro práci s VFK

V současné době existuje několik nástrojů pro práci s katastrálními daty. Ať už se jedná o moduly, které jsou distribuovány zdarma, nebo za úplaty, jako nadstavba pro komerční nástroje, případně o moduly dostupné pro open-source nástroje. S trochou nadsázky se dá říci, že většina programů pro práci s prostorovými daty má svůj zásuvný modul (nadstavbu) pro zpracování souborů VFK.

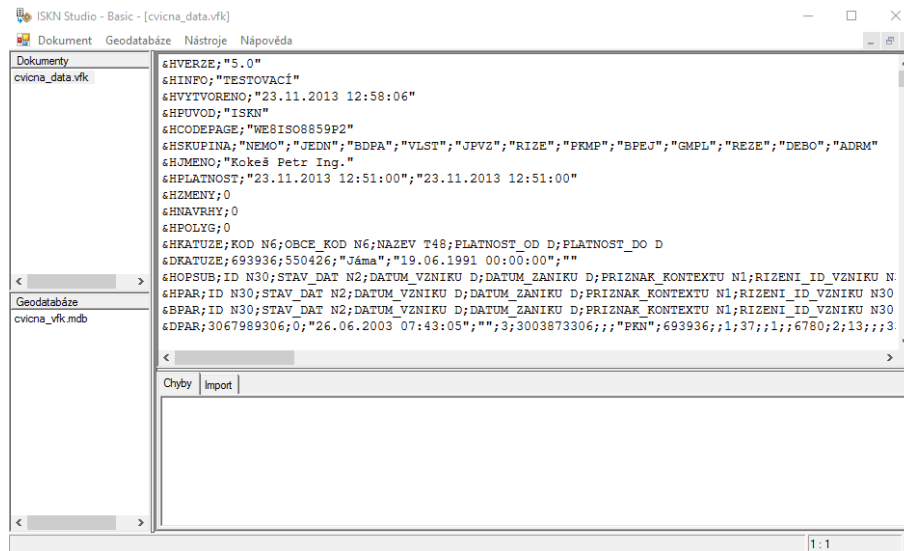
V kapitole se budeme věnovat vybraným nástrojům pracujícím s těmito daty. Moduly, které jsou šířeny zdarma a bylo je možné vyzkoušet, byly otestovány přímo. Popis komerčních modulů vychází především z dostupné dokumentace na příslušných oficiálních webových stránkách.

1.1 ISKN Studio pro ArcGIS

Aplikace ISKN Studio slouží pro převod dat VFK do ESRI geodatabáze, se kterou je schopen pracovat program ArcGIS. Struktura geodatabáze je navržena dle připravené šablony. Tato šablona je vždy vázána k dané verzi formátu VFK a může být stažena přímo ze stránek společnosti Arcdata Praha, která je zároveň tvůrcem této aplikace. Data načtená touto aplikací mohou být exportována také ve formátu XML. Díky této aplikaci může být v geodatabázi sestavena geometrie vektorových prvků. [10]

Struktura vytvořené geodatabáze je shodná s datovými bloky obsaženými ve vstupním souboru VFK (pro každý datový blok je vytvořena jedna tabulka).

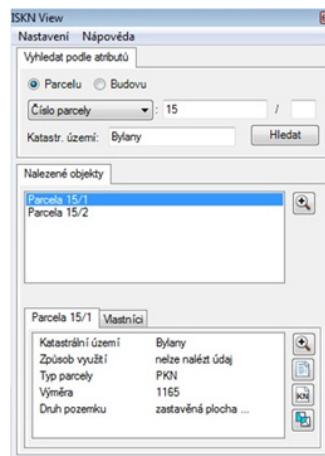
Aplikace disponuje kromě zpracování souboru VFK dalšími funkcionalitami. Jednou z hlavních je možnost kontroly struktury vstupního VFK souboru podle zvolené šablony nebo podle geodatabáze. Samozřejmostí je možnost uložení protokolu o zpracování. Aplikace je distribuována zdarma.



Obrázek 1.1: Aplikace ISKN Studio (zdroj: vlastní)

1.1.1 ISKN View pro ArcGIS

ISKN View je sesterskou aplikací ke zvýše zmíněnému ISKN Studio (1.1). Software je používán jako doplněk (*Add-In*) v programu ArcGIS verze 10.1 a vyšší.



Obrázek 1.2: Aplikace ISKN View (zdroj: [10])

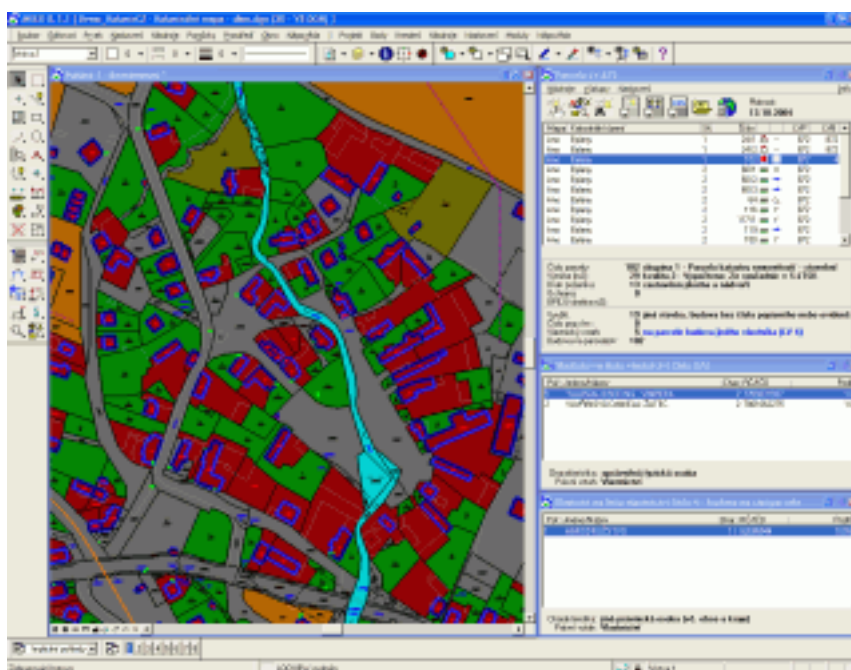
Díky ISKN View je umožněno rychlé a jednoduché vyhledávání v datech ISKN, která byla pomocí aplikace ISKN Studio převedena do ESRI geodatabáze. Aplikace je rovněž šířena bezúplatně. [10]

1.2 Import VFK, VKM

Jedná se o modul vytvořený společností GISOFT, který slouží k převodu a načtení dat ve formátech VFK a VKM do formátu DGN. Modul spolupracuje s produkty společnosti Bentley Systems, především MicroStation. Umožňuje načtení dat jak ve starém, tak i v novém výměnném formátu (viz kapitola č. 4) KN. Je dostupný jako volitelný modul pro nadstavby **MGEO**¹ a **SPIDER**². Může být použit v následujících případech:

Samostatně: Použití samostatně se hodí v případě, kdy jsou načítána data výměnného formátu obsahující pouze katastrální mapu. V tomto případě bude vstupní soubor převeden do podoby výkresu ve formátu DGN.

Spolu s modulem Práce s popisnými informacemi KN: Tento mód je užitečný v případě, kdy jsou v souboru spolu s katastrální mapou dostupné i popisné informace (případně jsou uvedeny pouze popisné informace).



Obrázek 1.3: Import dat KN ve výměnném formátu – ukázka použití (zdroj: [8])

Veškeré informace uvedené o tomto modulu vycházejí z oficiálního popisu modulu uvedeného na stránkách společnosti GISOFT. [8]

¹<http://www.gisoft.cz/MGEO/MGEO>

²<http://www.gisoft.cz/SPIDER/SPIDER>

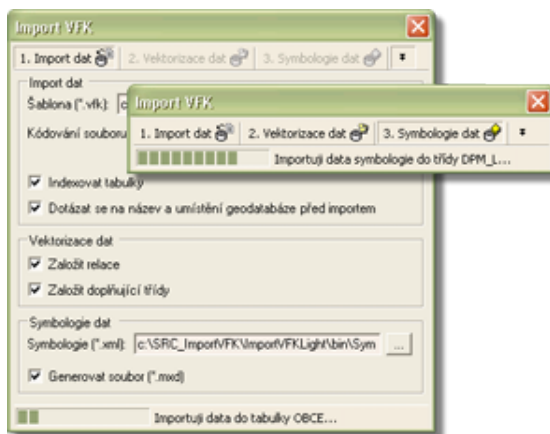
1.3 Spirit VFK

Software Spirit VFK je vytvořen a distribuován společností GEOREAL jako samostatně spustitelná desktopová aplikace. Slouží pro převod dat (VFK) katastru nemovitostí do jakékoli geodatabáze podporované společností ESRI.

Do geodatabáze jsou postupně importovány tabulky, relace a ostatní obsažené objekty ISKN. Takto vytvořená databáze může být použita v souvisejících aplikačních nadstavbách **Spirit KN³** a **Spirit Portál - KN⁴**, případně může sloužit pro další analytické práce nad daty KN. Import dat do geodatabáze probíhá v následujících krocích:

1. příprava geodatabáze (tvorba tabulek, relací, ...),
2. import dat VFK,
3. vektorizace parcel, budov a ostatních mapových vrstev,
4. optimalizace mapových vrstev, tvorba symbologie.

Symbologie (ve formátech MXD a LYR) vytvářená v posledním kroku se používá pro zobrazování dat ve výše zmíněných aplikačních nadstavbách pro ArcMap. Aplikace Spirit VFK může být využívána pro pravidelnou aktualizaci datových skladů katastru nemovitostí.



Obrázek 1.4: Spirit VFK – ukázka aplikace (zdroj: [22])

Pro aplikaci existuje i její odlehčená verze Spirit VFK Light, díky které je možné importovat data VFK do osobní geodatabáze ArcGIS (MS Access) nebo databáze MS SQL Server. Používání obou aplikací nevyžaduje znalost struktury dat ISKN nebo výměnného formátu VFK.

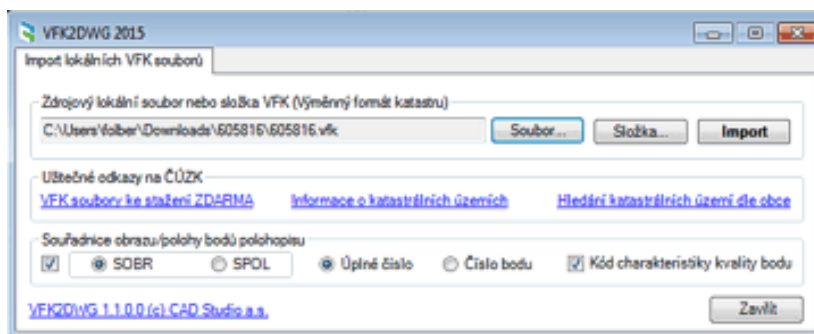
Veškeré výše uvedené informace pocházejí z oficiálních stránek produktu, viz [22].

³<http://www.georeal.cz/cz/spirit-desktop/spirit-kg>

⁴<http://www.georeal.cz/cz/spirit-server/portal-kg>

1.4 VFK2DWG

Jedná se o aplikaci od společnosti CAD Studio. Slouží jako nadstavba (utilita) pro produkty firmy Autodesk založených na AutoCadu (AutoCAD, AutoCAD Architecture, AutoCAD Map 3D, AutoCAD Civil 3D, ...). Díky této aplikaci je možné do výše uvedených programů načíst data VF ISKN (VFK) a dále s nimi pracovat.



Obrázek 1.5: Aplikace VFK2DWG (zdroj: [3])

Aplikace převádí *VFK* soubory na objekty (hranice parcel, parcelní čísla, vnitřní kresby, popisy, ...), se kterými je AutoCAD schopen pracovat. Tyto objekty jsou pomocí hypertextových odkazů provázány se stránkami ČÚZK (respektive s aplikací *Nahlížení do KN*), kde o nich mohou být zjištěny dodatečné informace.

Nejnovější verze aplikace pracuje i s daty formátu ve verzi 5.1 a je podporována i v AutoCAD 2016. Bohužel se jedná o komerční aplikaci a proto nemohla být otestována. Veškeré informace byly převzaty z oficiálních stránek společnosti CAD Studio, viz [3].

1.5 VFK2DB

VFK2DB je databázová varianta výše zmíněné aplikace (1.4), která se chová jako samostatně spustitelný program nezávislý na konkrétním programu GIS či CAD.

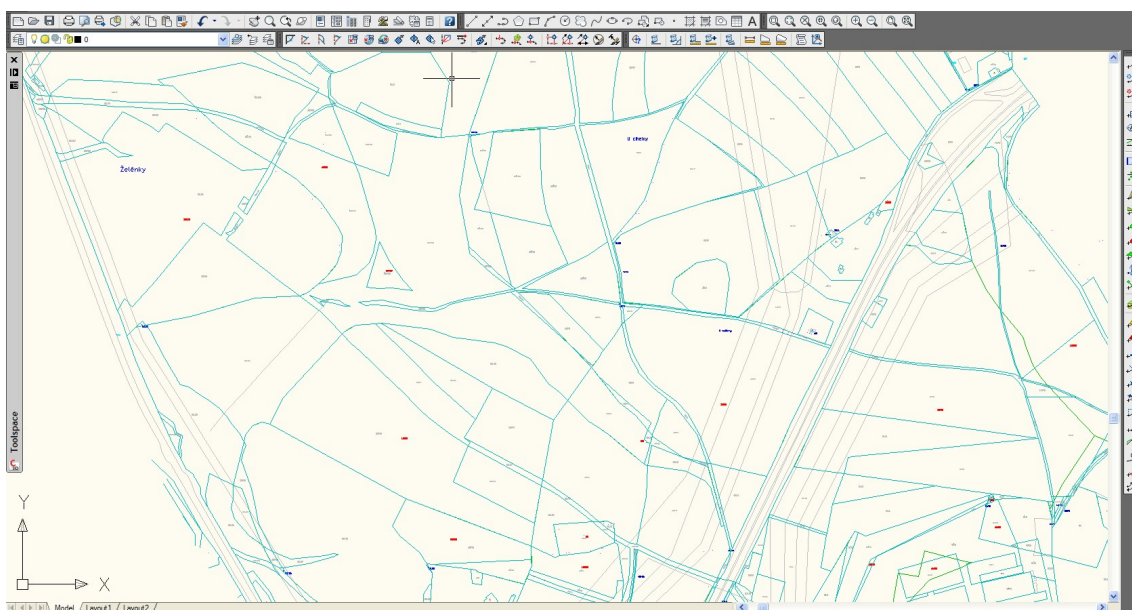
Aplikace importuje data z formátu *VFK* do relační databáze Oracle nebo MS SQL Server (v budoucnu se počítá s doplněním exportu do dalších typů databází, např. PostGIS, SQLite). Takto vytvořená databáze může být načtena některým z GIS produktů založených na prostorových SQL databázích (AutoCAD Map 3D, ESRI, Bentley, Intergraph, GeoServer, MapServer, ...).

Opět se jedná o komerční aplikaci společnosti CAD Studio, a proto nemohla být otestována. Veškerý zde uvedený popis vychází z oficiální dokumentace na stránkách společnosti, viz [2].

1.6 Topol VFK Import

Jak už ze samotného názvu plyne, aplikace Topol VFK Import byla vyvinuta společností Data System s.r.o. ve spolupráci se společností Topol Software. Aplikace disponuje vlastním grafickým prostředím, ve kterém je možné VFK data exportovat do formátů DWG a DXF, případně do vlastního formátu (OpenGIS MDB) společnosti Topol.

Opět se jedná o komerční aplikaci, a proto nemohla být vykoušena. Veškeré informace pocházejí z webových stránek výrobce, viz [25].



Obrázek 1.6: Topol VFK Import – ukázka zpracovaných dat (zdroj: [25])

1.7 GDAL – VFK Driver

VFK Driver, díky kterému je možné data VFK číst, je součástí knihovny GDAL (viz 2.2) od verze 1.7. Vstupní VFK soubor je knihovnou GDAL rozeznán jako OGR datasource, každý datový blok je poté vnímán jako OGR layer. Od GDAL verze 1.10 je podpora VFK přidána pouze v případě, kdy je knihovna kompilována s podporou SQLite (`./configure --with-sqlite`).

Driver si interně data při prvním čtení ukládá do databáze SQLite ve stejném adresáři, jako je umístěn VFK soubor. Při opětovném čtení driver používá pro čtení již vytvořenou databázi. Tímto se opakované načítání dat několikanásobně urychlí, viz porovnání níže. Implicitní chování driveru může být ovlivněno zadáním proměnných prostředí.

<pre> 1 # první čtení 2 real 0m11.547s 3 user 0m11.016s 4 sys 0m0.232s </pre>	<pre> 1 # opakované čtení 2 real 0m0.317s 3 user 0m0.284s 4 sys 0m0.028s </pre>
---	---

Jednou z nejdůležitějších je proměnná `OGR_VFK_DB_NAME` sloužící pro definici jména SQLite databáze. Neméně důležitá proměnná `OGR_VFK_DB_OVERWRITE` říká, že při každém čtení souboru VFK se vytváří databáze SQLite znovu, čtení tedy probíhá pouze ze souboru VFK. Níže je uvedena ukázka otevření souboru VFK. [7] [13]

```

1 $ ogrinfo Export_vse.vfk
2 Had to open data source read-only.
3 INFO: Open of 'Export_vse.vfk'
4     using driver 'VFK' successful.
5 1: PAR (Polygon)
6 2: BUD (Polygon)
7 3: ZPOCHN (None)
8     :
9 74: BUDOBJ (None)
10 75: ADROBJ (None)
  
```



Obrázek 1.7: Ukázka načtení VFK pomocí VFK Driveru GDAL v programu GRASS GIS (zdroj: vlastní)

Kapitola 2

Použité technologie

2.1 QGIS

QGIS je geografický informační systém, který je distribuován jako open-source¹ pod licencí *GNU General Public License*. Je oficiálním a klíčovým produktem organizace OSGeo. Díky přenositelnosti zdrojového kódu je použitelný na širokém spektru platform, ať už jsou to desktopové platformy Linux, MacOS, Windows, nebo mobilní platforma Android.



Obrázek 2.1: QGIS – logo (zdroj: [20])

Program umožňuje prohlížení, tvorbu a editaci velkého množství vektorových (Esri Shapefile, GeoJSON, GPX, . . .), ale i rastrových (GeoTIFF, JPEG, . . .) nebo databázových (PostGIS) formátů. Podporuje zpracování dat GPS a tvorbu mapových výstupů. Mimo jiné umožňuje provádět prostorové analýzy, analýzy terénu nebo analýzy síťové, práci s mapovou algebrou a mnoho dalšího.

QGIS nedisponuje tak širokou paletou nástrojů, jako jeho open-source kolega GRASS GIS. Jeho funkcionality ale může být rozšířena díky nepřebornému množství zásuvných modulů. Jedním z nejdůležitějších modulů pro analýzu geografických dat je zásuvný modul GRASS GIS, který zpřístupňuje funkce stejnojmenného programu. QGIS poté může sloužit jako jeho nadstavba. [20] [21]

¹Open-source software je takový software, k němuž zákazník dostane od jeho tvůrce zdrojový kód a může jej dále upravovat. Jednotlivé definice termínu „open source“ se liší zvláště v podmínkách pro další distribuci softwaru.[1]

2.2 GDAL

GDAL je knihovna určená pro čtení a zápis rastrových GIS formátů. Knihovna je vyvíjena pod hlavičkou Open Source Geospatial Foundation a vydávána pod licencí *X/MIT*. Knihovna používá jednoduchý abstraktní datový model pro všechny podporované datové formáty. Kromě toho nabízí také řadu užitečných nástrojů pro příkazovou řádku určených pro konverzi a zpracování dat.



Obrázek 2.2: GDAL – logo (zdroj: [5])

GDAL byla původně vyvíjena Frankem Warmerdamem a to do verze 1.3.2, posléze byla knihovna převedena na GDAL Project Management Committee, která je součástí Open Source Geospatial Foundation.

Knihovna OGR, která je od verze 2.0 součástí knihovny GDAL, slouží pro práci s daty ve vektorovém formátu.

GDAL je považován za jeden z hlavních open-source projektů. Knihovna je hojně využívána také v komerční GIS sféře. Knihovna je otevřená a poskytuje základní funkcionalitu potřebnou pro denní práci s rozsáhlým množstvím GIS formátů.[6] [5] [15]

2.3 Python

Jazyk Python je objektově orientovaný programovací jazyk, který efektivně používá víceúrovňové datové typy. Jedná se o jazyk interpretovaný, čímž se jeví jako ideální nástroj pro psaní skriptů, ale i rychlý vývoj aplikací. Je vyvíjen jako open-source software, díky čemuž se stává použitelným na velkém množství platforem (Linux, Windows, MacOS, ...). Jazyk je rozšiřitelný o široké spektrum modulů, které umožňují řešit problematiku takřka z jakékoli oblasti. V současné době je Python vyvíjen ve dvou verzích, ve verzi 2.x a v novější verzi 3.x. [16] [19]



Obrázek 2.3: Python – logo (zdroj: [19])

2.4 PyQt

PyQt je modul, který zpřístupňuje knihovnu Qt pro programovací jazyk Python. Spolu s PySide se jedná o nejnámější a nejpoužívanější modul pro Python postavený nad knihovnou Qt. Je vyvíjen britskou firmou Riverbank Computing ve dvou verzích. Ve verzi 4, podporující knihovnu Qt 4, a ve verzi 5, která podporuje novější verzi Qt knihovny. Modul je dostupný na všech platformách, které podporují knihovnu Qt (Windows, MacOS/X a Linux). PyQt je šířeno pod tzv. dvojí licencí, *GNU GPL v3* a *Riverbank Commercial License*. Spolu s těmito licencemi je dostupné i pod komerční licencí.



Obrázek 2.4: PyQt – logo (zdroj: [18])

Pro grafický návrh aplikace je vhodné použít nativní grafické uživatelské rozhraní Qt Designer. Výstupem z tohoto programu je soubor obsahující vzhled aplikace ve formátu *XML*. PyQt je poté schopné tento formát převést do kódu jazyka Python. Pro komunikaci mezi objekty je využíváno signálů a slotů, díky čemuž je vytvoření komponent velice snadné.

PyQt v sobě kombinuje mocnost knihovny Qt s jednoduchostí jazyka Python, což z něj dělá výkonný nástroj pro vývoj grafických aplikací. [17] [18] [24]

Kapitola 3

Informační systém katastru nemovitostí

ISKN je integrovaný informační systém pro podporu výkonu státní správy katastru nemovitostí a pro zajištění jeho uživatelských služeb. Obsahuje prostředky pro současné vedení souborů popisných informací (SPI) a souborů geodetických informací (SGI). Dále jsou v něm obsaženy prostředky pro podporu správních a administrativních činností při vedení katastru nemovitostí a pro správu dokumentačních fondů.

Informace obsažené v této kapitole jsou čerpány především z popisu Informačního systému katastru nemovitostí na stránkách ČUZK, viz [9] a článku v časopise Geoinformatics FCE CTU ([11]).



Obrázek 3.1: ČUZK – logo (zdroj: [9])

3.1 Historie a vývoj

Vývoj systému byl započat v roce 1997 ve spolupráci se společností APP Czech s.r.o.¹, která fungovala jako systémový integrátor a dodavatel aplikačního programového vybavení pro ČUZK. Dalšími společnostmi podílejícími se na vývoji ISKN byly Infinity, a.s., Compaq Computer s.r.o.², Oracle Czech, s.r.o., Bentley Systems, s.r.o., BEA Systems, s.r.o.

¹Dnes společnost funguje pod názvem NESS Czech s.r.o.

²Dnes pod názvem HP.

Systém byl nasazen do provozu v září roku 2001, a to na všech katastrálních pracovištích včetně centrály. Doladování a převzetí závěrečných etap probíhalo v roce 2002. V téže roce byl dokončen audit systému.

Implementace ISKN plně nahradila dřívější způsob vedení katastru nemovitostí. ISKN integroval vedení a správu katastru nemovitostí pod jediný informační systém společný pro všechna pracoviště katastrálních úřadů a centrum. To vedlo k tomu, že je možné zveřejňovat a poskytovat aktuální data z katastru nemovitostí prostřednictvím dálkového přístupu během několika málo minut, a to z celého území republiky.

Data jsou do systému ISKN ukládána pomocí Spatial Cartridge Option do databáze Oracle. Podpora vzdáleného přístupu k datům pomocí sítě Internet je zajištěna pomocí BEA WebLogic. Systémový management využívá nástrojů CA Unicenter.

V roce 2004 byla uzavřena nová smlouva se společností NESS Czech s.r.o. na rozvoj a údržbu informačního systému v letech 2004 – 2006. V tomto období byl zmodernizován především Dálkový přístup do katastru nemovitostí a zavedena orientační mapa parcel. Důležitou inovací bylo zavedení elektronické značky pro výpis z katastru nemovitostí a pro kopii katastrální mapy ³.

Společnost NESS Czech s.r.o. poté v dalších letech vyhrála několik veřejných zakázek týkajících se údržby a rozvoje ISKN. Hlavním cílem bylo převedení decentralizovaného systému (107 lokálních databází replikovaných do centrální databáze) na centralizovaný systém, ve kterém byla data ISKN uložena pouze v jedné databázi. Spolu s touto úpravou byla změněna i architektura z původní client/server na třívrstvou architekturu. Architektura je postavena na platformě Oracle Forms/Reports 10g a databázi Oracle 10g. Další změnou byl přechod na vyšší verzi Bentley nástroje pro správu prostorových dat.

ISKN byl nadále zlepšován. Za zmínku stojí především systém pro Dálkový přístup do katastru nemovitostí nebo zavedení možnosti získat informaci o ukončení řízení pomocí SMS nebo e-mailové zprávy.

3.2 Hlavní charakteristiky ISKN

3.2.1 Optimalizace uložení dat

Díky zvolení jednotného datového modelu pro uložení popisných a prostorových dat v databázi Oracle spolu s daty týkajícími se správních řízení byla umožněna současná aktualizace popisných a prostorových dat a udržení jejich vzájemné konzistence. Pro optimalizaci byla také přijata koncepce samostatné evidence budov

³Tento krok umožnil, aby tzv „ověřující“ podle zákona č. 365/2000 Sb., o informačních systémech veřejné správy, v platném znění, mohli poskytovat ověřené výpisy z katastru nemovitostí, převedené z elektronické do listinné podoby.[9]

a bežešvé digitální katastrální mapy. Od konce roku 2001 jsou uchovávány také veškerá historická data popisných a prostorových dat, díky čemuž je možné sestavovat data do potřebných výstupů k historickému datu od zavedení ISKN v roce 2001.

3.2.2 Optimalizace procesů při správě KN

Do systému ISKN byla zavedena celá řada automatických kontrol pro proces zapsání změny do KN. Dále bylo umožněno převzetí aktuálních dat z jiných registrů (např. registr obyvatel) a ostatních informačních systémů. Postup provedení změny dat KN je následující: na základě návrhu je připraven budoucí stav, který je možné před jeho zplatněním zobrazit (SPI, SGI), případně v něm provádět úpravy. Toto zajišťuje důkladnou kontrolu výsledného stavu katastru. Proces realizace změny je navíc zajištěn i technicko-organizačními opatřeními (návrh změny a kontrolu, včetně zplatnění provádí vždy jiná osoba dle přidělených uživatelských rolí).

Díky novým procesům ve zpracování dat/návrhů změn je možné částečné nabytí platnosti geometrického plánu s automatizovanou změnou návrhu změny v budoucím stavu. Nové procesy také umožňují aktualizaci dat katastru nemovitostí takovým způsobem, aniž by zamkly aktualizovaná data. Pouze se jimi řeší konflikty v aktualizaci stejných dat.

Součástí ISKN je také jednotná centrální správa číselníků, která vnáší jednotnost do procesu zpracování změn na katastrálních úřadech. Tímto se rapidně zvyšuje konzistence a kvalita datové základny. Některé z centrálních číselníků nebo seznamů jsou přebírány z externích datových zdrojů (např. číselníky územní identifikace, PSČ).

3.2.3 Bezpečnost

Vysoká bezpečnost ochrany dat je zajištěna kombinací hardwarových prostředků s operačním systémem, databází a vlastní aplikací ISKN. Nepřetržitý provoz je zajištěn pomocí technologie databázových a aplikačních clusterů a tím, že je celá infrastruktura zdvojená (primární a záložní centrum). Do záložního centra jsou replikována veškerá data tak, aby byl v případě náhlého výpadku primárního centra zajištěn nepřetržitý provoz ISKN.

3.3 Poskytování dat

Poskytování dat je umožněno na základě vyhlášky číslo 358/2013 Sb., o poskytování údajů z katastru nemovitostí.

3.3.1 Poskytování dat dálkovým přístupem

Na základě registrace je umožněno poskytování dat (zdarma, nebo za úplatu podle typu zákazníka) prostřednictvím sítě Internet. Výpisy z KN a snímky katastrální mapy mají povahu veřejných elektronických listin (jsou opatřeny elektronickou značkou) a mohou být převedeny do podoby listinných veřejných listin. Tímto způsobem je v současné době vyřizována více než třetina výstupů.

Více informací o této metodě poskytování dat je spolu s aplikací dostupných na stránkách ČUZK (<http://www.cuzk.cz/aplikace-dp/>).

3.3.2 Poskytování dat ve výměnném formátu ISKN

Data z KN mohou být poskytována v textovém souboru, který obsahuje záznamy v pevně definované struktuře. Více informací o tomto výměnném formátu je uvedeno v kapitole č. 4.

Kapitola 4

Výměnný formát ISKN

V této kapitole je ve stručnosti popsána historie vývoje výměnného formátu ISKN spolu s jeho popisem, ve kterém se věnuji především sekcím důležitým pro vývoj zásuvného modulu pro QGIS, který dokáže data v tomto formátu zobrazit. Kapitola čerpá především z oficiální dokumentace výměnného formátu, viz [23].

4.1 Vývoj formátu

Hlavním milníkem ve vývoji výměnného formátu bylo zavedení ISKN, viz. kapitola č. 3. Do této doby byly soubory SPI a SGI ukládány odděleně, což se právě se zavedením ISKN změnilo. Tento krok vedl k vytvoření nového výměnného formátu (NVF), který postupně nahrazoval starý výměnný formát (SVF). [14]

4.1.1 Výměnný formát KN před ISKN

Tento formát je po zavedení nového formátu také nazýván jako *starý výměnný formát* – **SVF**. Byl vytvořen roku 1996, kdy začala vznikat digitalizace SGI. Obsahuje dvě samostatné části:

1. **SPI** – Obsahuje informace o vlastnících, parcelách a nabývacích titulech. Byl distribuován ve dvou formátech:
 - (a) soubory ve formátu *.dbf*: Tento typ souboru byl dále dělen na další dvě části:
 - i. SPI bez jiných právních vztahů (bez JPV),
 - ii. SPI s jinými právními vztahy (s JPV).
 - (b) soubory ve formátu *.txt*: SB, SC, SE

Data byla poskytována v následujících rozsazích: podle územní jednotky (katastrální území, obec, okres, ČR), dle výběru parcel, nebo na základě oprávněného subjektu (pouze ve formátu *.txt*).

Ve výše zmíněných formátech (*.dbf*, *.txt*) jsou značné nesoulady. Ty jsou způsobeny hlavně neexistencí některých položek v novém datovém modelu ISKN nebo jejich rozdílnou interpretací.

2. SGI – Jsou v něm obsaženy informace o poloze nemovitostí.

Data byla poskytována pro katastrální území, kde již byla provedena digitalizace.

V současné době je již oficiální podpora ukončena a byl nahrazen právě novým výměnným formátem. [28]

4.1.2 Výměnný formát VF ISKN

Formát je nazýván také jako *nový výměnný formát – NVF*. V tomto formátu jsou obsaženy zároveň popisné i grafické informace včetně dat o řízení. Data jsou vytvářena ve dvou stavech:

- **Stavová data** – Data jsou vygenerována vzhledem ke konkrétnímu časovému okamžiku. Obsahují vždy kompletní data pro daný okamžik. Práce s těmito daty je řešena v původní verzi zásuvného modulu pro QGIS.
- **Změnová data** – Jsou v nich obsaženy pouze změny za požadovaný časový úsek. Zpracováním a zobrazením změnových dat v programu QGIS se zabývá právě tato diplomová práce.

Data jsou poskytována v následujících rozsazích:

- územní jednotka (katastrální území, obec, okres, ČR),
- oprávněný subjekt,
- výběr parcel,
- výběr parcel polygonem v mapě.

Do výměnného souboru je možné dle přání zákazníka vybrat libovolné kombinace datových skupin, viz tab. 4.1. [14]

Tabulka 4.1: Datové skupiny VF ISKN (zdroj: [27])

Název skupiny	Obsah
Nemovitosti	parcely a budovy
Jednotky	bytové jednotky
Bonitní díly parcel	kódy BPEJ k parcelám
Vlastnictví	listy vlastnictví, oprávněné subjekty a vlastnické vztahy
Jiné právní vztahy	ostatní právní vztahy kromě vlastnictví
Řízení	údaje o řízení (vklad, záznam, ...) a listiny
Prvky katastrální mapy	katastrální mapy v digitální podobě
BPEJ	hranice BPEJ včetně kódů
Geometrický plán	geometrické plány
Rezervovaná čísla	rezervovaná parcelní čísla a čísla PBPP
Definiční body	definiční body parcel a staveb
Adresní místa	adresní místa budov

4.2 Struktura výměnného formátu ISKN

Tato kapitola pojednává o struktuře výměnného formátu ISKN. Nejsou zde popsány a do detailu rozvedeny veškeré datové bloky formátu, ale pouze ty nejdůležitější prvky formátu vzhledem k zásuvnému modulu pro QGIS. Podrobný popis formátu je dostupný v oficiální dokumentaci (viz [23]), ze které tato kapitola čerpá. Veškeré ukázky výměnného formátu jsou pořizeny z testovacích dat dostupných na stránkách ČUZK (viz [27]).

Datový soubor VFK se skládá ze tří základních částí, které budou samostatně popsány v následujícím textu této kapitoly:

- hlavička **&H**,
- datové bloky **&B**,
- koncový znak **&K**.

Datový soubor je vytvářen v kódování češtiny dle ČSN ISO 8859-2 (ISO Latin2)¹. Desetinným oddělovačem je tečka (.). Datum a čas je uveden ve tvaru "03.06.1999 09:58:42". Jednotlivé záznamy na řádcích jsou odděleny pomocí středníku (;). Každá datová věta je ukončena pomocí souslednosti znaků **<CR><LF>**. Znak „**Q**“ znamená, že následující řádek souboru výměnného formátu je pokračováním předchozího řádku a tvoří jedinou datovou větu, která v textové položce obsahuje formátovací znaky **<CR><LF>**.

¹Ve výjimečných případech je možné použít kódování WIN1250. Toto kódování je použito i v souboru ve formátu XML verze 1.0.

4.2.1 Hlavička &H

Každý řádek hlavičky začíná sousledností znaků &H, po které následuje označení položky, např. **VERZE**. Jednotlivé údaje jsou odděleny pomocí středníku. Hlavička obsahuje několik povinných řádků, jejichž seznam je uveden v tabulce 4.2.

Tabulka 4.2: Seznam položek hlavičky (zdroj: [23])

Položka	Popis
VERZE	označení verze VF
VYTVORENO	datum a čas vytvoření souboru
PUVOD	původ dat
CODEPAGE	označení kódové stránky
SKUPINA	seznam skupin datových bloků souboru
JMENO	jméno osoby, která soubor vytvořila
PLATNOST	časová podmínka použitá pro vytvoření souboru
ZMENY	stavová, nebo změnová data
POLYG	omezující podmínka – polygon
KATUZE	omezující podmínka – katastrální území
OPSUB	omezující podmínka – oprávněné subjekty
PAR	omezující podmínka – parcely

Příklad prvních řádků hlavičky je uveden v tabulce 4.3. Tabulka byla vytvořena na základě testovacích dat a obsahuje i některé nepovinné položky, např. &HINFO.

Tabulka 4.3: Ukázka hlavičky (zdroj: [23])

Položka	Atributy
&HVERZE	"5.0"
&HINFO	"TESTOVACÍ"
&HVYTVORENO	"23.11.2013 12:58:06"
&HPUVOD	"ISKN"
&HCODEPAGE	"WE8ISO8859P2"
&HSKUPINA	"NEMO";"JEDN";"BDPA";"VLST";"JPVZ"
&HJMENO	"Kokeš Petr Ing."
&HPLATNOST	"23.11.2013 12:51:00";"23.11.2013 12:51:00"
&HZMENY	0
&HNAVRHY	0
&HPOLYG	0

VERZE: Pouze jeden řádek označující verzi souboru VFK.

VYTVOŘENO: Datum a čas, kdy byl datový soubor vygenerován.



PŮVOD: Specifikuje původ dat. Standardně je zde uvedeno „ISKN“.

CODEPAGE: Označení kódové stránky. Hodnota „WE8ISO8859P2“ značí kódování češtiny dle ČSN ISO 8859-2. Hodnota „EE8MSWIN1250“ slouží pro označení kódování češtiny dle MS WIN1250.

SKUPINA: Uvádí se zde seznam datových bloků souboru. Např. `&HSKUPINA; "Zkratka skupiny\;[\Zkratka skupiny" ...]`.

JMÉNO: Jméno osoby, která soubor vytvořila. Např. `&HJMENO; "Jméno Příjmení"`.

PLATNOST: Časová podmínka použitá pro vytvoření souboru. Zde jsou možné dvě varianty:

- Data jsou platná v daném čase. `&HPLATNOST; "03.12.2013 09:56:42"; "03.12.2013 09:56:42"`,
- data jsou platná v daném období. `&HPLATNOST; "03.12.2012 09:56:42"; "03.12.2013 09:56:42"`.

S tímto souvisí položka `&HZMENY`, která nabývá hodnot 0/1 a označuje, zda se jedná o data stavová, nebo změnová. Položka `&HNAVRHY` nabývá také hodnot 0/1 a značí, zda jsou v souboru obsaženy potvrzené geometrické plány, či nikoliv.

KATUZE: Obsahuje jeden řádek, který popisuje hlavičku omezující podmínky katastrálních území. Další řádky začínající `&D` tvoří omezující podmínku. Počet datových řádků udává počet katastrálních území, která omezující podmínku tvoří. Pokud v omezující podmínce není žádné katastrální území, bude uvedena pouze hlavička. Pro ujasnění je zde uveden příklad z testovacích dat.

```
1 &HKATUZE;KOD N6;OBCE_KOD N6;NAZEV T48;PLATNOST_OD D;PLATNOST_DO
2 &DKATUZE;693936;550426;"Jama";"19.06.1991 00:00:00";""
```

OPSUB: První řádek popisuje hlavičku omezující podmínky oprávněných subjektů. Další řádky s daty poté omezující podmínku tvoří, obdobně jako je uvedeno u omezující podmínky pro katastrální území. Počet datových řádků je shodný s počtem oprávněných subjektů v omezující podmínce.

PAR: První řádek popisuje hlavičku omezující podmínky parcel. Další řádky tvoří omezující podmínku. Počet datových řádků je shodný s počtem parcel uvedených v omezující podmínce.

POLYG: Tento údaj může nabývat hodnot 0/1. Pokud je uvedena hodnota 1, tak je obsah souboru odvozen z polygonu. V takovém to případě musí být polygon na dalších řádcích definován svými vrcholy. Takto zadaný polygon může mít nejvýše 101 vrcholů. Příklad zadání omezujícího polygonu:



```

1 &HPOLYGDATA;675124.12;1024587.24
2 &HPOLYGDATA;675224.12;1024687.24
3 &HPOLYGDATA;675184.12;1024537.24

```

4.2.2 Datové bloky

Datové bloky obsahují řádky dvojího typu:

- uvozující řádek bloku **&B** – obsahuje seznam atributů s jejich datovými typy, viz tab. 4.4,
- datové řádky **&D** – v řádku jsou uvedeny vlastní data.

Tabulka 4.4: Datové typy ISKN (zdroj: [14])

Kód	Datový typ	Číslo za kódem
N	číselný	maximální délka položky
T	textový	maximální délka textu
D	datumový	ve tvaru DD.MM.YYYY HH:MI:SS

Níže je uveden příklad datového bloku pro blok „PARCELA“. Ukázka je pořízena z testovacích dat.

```

1 &BPAR;ID N30;STAV_DAT N2;DATUM_VZNIKU D;DATUM_ZANIKU
2 D;PRIZNAK_KONTEXTU N1;RIZENI_ID_VZNIKU N30;RIZENI_ID_ZANIKU
3 N30;PKN_ID N30;PAR_TYPE T10;KATUZE_KOD N6;KATUZE_KOD_PUV
4 N6;DRUH_CISLOVANI_PAR N1;KMENOVE_CISLO_PAR N5;ZDPAZE_KOD
5 N1;PODDELENI_CISLA_PAR N3;DIL_PARCELY N1;MAPLIS_KOD N30;
  ZPURVY_KOD
6 N1;DRUPOZ_KOD N2;ZPVYPA_KOD N4;TYP_PARCELYN1;VYMERA_PARCELY
7 N9;CENA_NEMOVITOSTI N14.2;DEFINICNI_BOD_PAR T100;TEL_ID N30;
  PAR_ID
8 N30;BUD_ID N30;IDENT_BUD T1;SOUCASTI T1;PS_ID N30;IDENT_PS T1
9
10 &DPAR;3067989306;0;"26.06.2003 07:43:05";"";3;3003873306;;; "PKN
  ";
11 693936;;;1;37;;;1;;;6780;2;13;;;332;;; "674674306;;323700306;"a";"n
  ";;"n"

```

Seznam skupin datových bloků ISKN

V této sekci je uveden popis jednotlivých skupin datových bloků. Jsou zde uvedeny pouze nejpodstatnější informace. Podrobný popis lze dohledat v oficiální dokumentaci formátu VFK (viz [23]), ze které tento text převážně vychází.

NEMOVITOSTI: Jedná se o největší skupinu datových bloků. Celkem jich může obsahovat až 21. V této skupině se nachází dva nejdůležitější bloky z pohledu zásuvného modulu pro QGIS, a to bloky PAR a BUD. Právě tyto dva bloky jsou pomocí zásuvného modulu vizualizovány. Dále je zde obsažen například číselník způsobů využití pozemku nebo způsobů využití budov. Seznam všech bloků v této skupině je uveden v tabulce 4.5. Znak (*) uvedený v tabulce znamená, že daný blok nepodléhá historizaci.

Tabulka 4.5: Seznam datových bloků ve skupině „NEMOVITOSTI“ (zdroj: [23])

Kód	Popis
PAR	Parcely
BUD	Budovy
CABU	Části budov
ZPOCHN*	Číselník způsobů ochrany nemovitosti
DRUPOZ*	Číselník druhů pozemku
ZPVYPO*	Číselník způsobů využití pozemku
ZDPAZE*	Číselník zdrojů parcel ZE
ZPURVY*	Číselník způsobů určení výměry
TYPBUD*	Číselník typů budov
MAPLIS*	Číselník mapových listů
KATUZE*	Číselník katastrálních území
OBCE*	Číselník obcí – vázaně
CASOBC*	Číselník částí obce – vázaně
OKRESY*	Číselník okresů – vázaně
KRAJE*	Číselník krajů – vázaně
NKRAJE*	Číselník nových krajů – vázaně
RZO	Přiřazení způsobu ochrany k nemovitostem
ZPVYBU*	Způsob využití budov
PS	Práva stavby
RU	Přiřazení účelu práva stavby
UCEL	Číselník účelů práva stavby

JEDNOTKY: V této skupině jsou uvedeny bytové či nebytové prostory, které byly označeny příslušnou listinou jako jednotka. Pro každou jednotku je uveden její popis (jednoznačně ji identifikuje v rámci budovy), typ a způsob využití. Ke ka-

ždé jednotce je dále uveden spoluvlastnický podíl ($\frac{\text{velikost podlahové plochy}}{\text{celková plocha všech jednotek v domě}}$). [14]

Tabulka 4.6: Seznam datových bloků ve skupině „JEDNOTKY“ (zdroj: [23])

Kód	Popis
JED	Jednotky
TYPJED*	Číselník typů jednotek
ZPVYJE*	Způsob využití jednotek

BONITNÍ DÍLY PARCEL: Jsou zde uvedeny informace o bonitních dílech parcely. Ve skupině se nachází pouze jeden datový blok (BDP), ve kterém je popsán vztah mezi BPEJ² a parcelou. [14]

VLASTNICTVÍ: Tato skupina bloků obsahuje informace o vlastnictví. Jako vlastník zde může být uvedena fyzická osoba, právnická osoba nebo jiný oprávněný uživatel (manželé v bezpodílovém spoluvlastnictví). Ve skupině se může nacházet několik datových bloků, viz tab. 4.7. [14]

Tabulka 4.7: Seznam datových bloků ve skupině „VLASTNICTVÍ“ (zdroj: [23])

Kód	Popis
OPSUB	Oprávněné subjekty
VLA	Vlastnictví
CHAROS*	Číselník charakteristik oprávněných subjektů
TEL	Katastrální tělesa

JINÉ PRÁVNÍ VZTAHY: Obsahuje informace o jiných než vlastnických vztazích jednoho oprávněného subjektu (nemovitosti) ke konkrétnímu předmětu (nemovitosti, vlastnictví, dalšímu jinému právnímu vztahu). Seznam a popis datových bloků je uveden v tabulce č. 4.8. [14]

ŘÍZENÍ: Tato skupina je druhou nejobsáhlejší skupinou ve výměnném formátu. Může být obsažena ve změnovém exportu z ISKN – v tomto případě obsahuje pouze záznamy, které byly v daném časovém intervalu změněny. Obsahuje několik datových bloků, viz tab. 4.9. [14]

²**Bonitovaná půdně-ekologická jednotka:** Základní určovací a oceňovací jednotka produkční schopnosti zemědělské půdy. Je vyjádřena číselným kódem – číslice kódu vyjadřují půdně-klimatické vlastnosti půdy. Jednotky tvoří ohraničený územní celek, který má specifické ekologické vlastnosti a bioenergetický potenciál. [26]

Tabulka 4.8: Seznam datových bloků ve skupině „JINÉ PRÁVNÍ VZTAHY“ (zdroj: [23])

Kód	Popis
JPV	Jiné právní vztahy
TYPRAV*	Číselník typů právních vztahů
RJPV	Vazba JPV k jinému věcnému právu

Tabulka 4.9: Seznam datových bloků ve skupině „ŘÍZENÍ“ (zdroj: [23])

Kód	Popis
RIZENI*	Řízení (vklad, záznam)
RIZKU*	Vazba Řízení – Katastrální území
OBJRIZ*	Objekty řízení (parcely, budovy, ...)
PRERIZ*	Předměty řízení
UCAST*	Účastníci řízení
ADRUC*	Adresy účastníků řízení
LISTIN*	Listiny
DUL*	Další údaje listin
LDU*	Vazba Listiny – Další údaje listin
TYPLIS*	Číselník typů listin
TYPPRE*	Číselník typů předmětu řízení
TYPRIZ*	Typy řízení
TYPUCA*	Typy účastníků řízení
UCTYP*	Vazba Účastníci – Typy účastníků řízení
RL	Přiřazení listin k nemovitostem, vlastnictví a jiným právním vztahům
OBESMF*	Obeslání účastníků řízení

PRVKY KATASTRÁLNÍ MAPY: Tato skupina je jednou z nejdůležitějších pro zásuvný modul, respektive pro driver VFK v knihovně GDAL. Jsou v ní totiž obsaženy jak popisné, tak hlavně polohopisné informace o prvcích polohopisu. Z nich (tedy hlavně z prvních dvou datových bloků SOBR a SBP) je tvořena geometrie vektorové mapy. Spolu s nimi skupina obsahuje další neméně důležité datové bloky, viz tab. 4.10. [14]

Tabulka 4.10: Seznam datových bloků ve skupině „PRVKY KATASTRÁLNÍ MAPY“ (zdroj: [23])

Kód	Popis
SOBR*	Souřadnice obrazu bodů polohopisu v mapě
SBP	Spojení bodů polohopisu – definuje polohopisné liniové prvky
SBM	Spojení bodů mapy – definuje nepolohopisné liniové prvky
KODCHB*	Číselník kódů charakteristiky kvality bodu
TYPPOS*	Číselník typů souřadnicových systémů
HP	Hranice parcel
OP	Obrazy parcel (parcelní číslo, značka druhu pozemku, ...)
OB	Obrazy budov (obvod budovy, značka druhu budovy)
DPM	Další prvky mapy
OBBP	Obrazy bodů BP
TYPPPD*	Číselník typů prvků prostorových dat
ZVB	Zobrazení věcných břemen
POM	Prvky orientační mapy
SPOM	Spojení prvků orientační mapy – definuje liniové prvky
SPOL	Souřadnice polohy bodů polohopisu (měřené)

BPEJ: Skupina BPEJ obsahuje dva datové bloky, viz tab. 4.11. Jsou v ní obsaženy informace o bonitovaných půdně ekologických jednotkách. BPEJ je základní mapovací a oceňovací jednotka zemědělských půd, která vyjadřuje rozdílné produkční a ekonomické efekty zemědělského území. Hranice BPEJ nejsou součástí katastrální mapy. Pouze tvoří rozhraní mezi dvěma jednotkami. [14]

Tabulka 4.11: Seznam datových bloků ve skupině „BPEJ“ (zdroj: [23])

Kód	Popis
HBPEJ	Hranice BPEJ
OBPEJ	Označení BPEJ

GEOMETRICKÝ PLÁN: Tato skupina obsahuje sadu bloků popisujících geometrický plán a hlavičku dalších změn v KM, které nejsou prováděny geometrickým plánem. Obsahuje několik datových bloků, jejichž seznam je uveden



v tabulce č. 4.12. Ve skupině je obsažena tabulka pro uchování záznamů podrobného měření změn jak v terénu, tak i změn, které s měřením v terénu nesouvisí (slučování parcel, demolice budov, ...). [14] [23]

Tabulka 4.12: Seznam datových bloků ve skupině „GEOMETRICKÝ PLÁN“ (zdroj: [23])

Kód	Popis
NZ	Hlavičky geometrických plánů a ostatních změn KM
ZPMZ	Hlavičky ZPMZ
NZZP	Vazební tabulka návrhy změn KM – ZPMZ
PARG	Parcely GP
BUDG	Budovy GP
BDPG	Bonitní díly parcel GP
HPG	Hranice parcel GP
OPG	Obrazy parcel GP
OBG	Obrazy budov GP
ZVBG	Zobrazení věcných břemen GP
DPMG	Další prvky mapy GP
SBPG	Spojení bodu polohopisu GP
OBPEJG	Označení BPEJ GP
SBMG	Spojení bodů mapy GP
HBPEJG	Hranice BPEJ GP
OBDEBOG	Obrazy definičních bodů parcel a budov GP

REZERVOVANÁ ČÍSLA: Rezervovanými čísly se v této skupině myslí parcelní čísla, která byla rezervována pro účely vyhotovení geometrického plánu. Před potvrzením geometrického plánu probíhá kontrola, jestli byla použita přidělená rezervovaná parcelní čísla. Při zápisu nové parcely do KN se její číslo z tabulky RECI maže. Úplné parcelní číslo musí být jedinečné v rámci tabulek PAR a RECI.

Datový blok DOCI obsahuje všechna parcelní čísla, která kdy byla použita za dobu elektronického vedení katastru nemovitostí v informačním systému ISKN (od r. 2001). Seznam všech datových bloků je uveden v tabulce č. 4.13. [14] [23]

DEFINIČNÍ BODY: Skupina obsahuje pouze jeden datový blok OBDEBO. V tomto bloku jsou obsaženy obrazy definičních bodů parcel, budov a částí budov (pokud jsou v ISKN naplněny). Jsou zde uvedeny údaje o souřadnicích a odkazech (ID) na objekty v KN. [23]

ADRESNÍ MÍSTA: Datový blok BUDOBJ zajišťuje vazbu mezi budovami a adresami pomocí ID budovy a kódu objektu. Tento blok nepracuje s historií –

Tabulka 4.13: Seznam datových bloků ve skupině „REZERVOVANÁ ČÍSLA“ (zdroj: [23])

Kód	Popis
RECI	Rezervovaná parcelní čísla
DOCI	Dotčená parcelní čísla
DOHICI	Dotčená historická parcelní čísla
REZBP	Rezervovaná čísla bodu PBPP

obsahuje vždy aktuální data bez ohledu na datum, ke kterému je export NVF vytvořen.

Ve druhém bloku (ADROBJ) jsou uvedeny odkazy na adresy budov, které jsou obsaženy v bloku nemovitostí. Blok opět nepracuje s historií. [23]

Tabulka 4.14: Seznam datových bloků ve skupině „ADRESNÍ MÍSTA“ (zdroj: [23])

Kód	Popis
BUDOBJ	Odkazy objektů na adresy
ADROBJ	Adresy

4.2.3 Koncový znak &K

Specifickou částí výměnného formátu je takzvaný „koncový znak“ &K. Načtení tohoto znaku signalizuje konec souboru výměnného formátu. Pro driver VFK v knihovně GDAL znak znamená konec načítání.

4.3 Změnové věty v NVF

Změnový export nelze provést nad všemi skupinami datových bloků. Proveden může být pouze nad následujícími:

- nemovitosti,
- jednotky,
- bonitní díly parcel,
- vlastnictví,
- JPV,
- řízení,
- prvky katastrální mapy,

- BPEJ.

Objekty, které byly vybrány v parametrech při spuštění exportu, jsou součástí datového souboru i v případě, kdy na nich nebyla provedena žádná změna (tzn. jsou platné). Pro ostatní objekty ve vybraných datových skupinách se exportují pouze změnové věty.

4.3.1 Obsah změnového exportu – Typy tabulek

Z pohledu exportu změnových vět rozlišujeme několik skupin tabulek, které jsou rozděleny podle toho, zda podléhají principu historizace, či nikoliv.

Tabulky předmětu KN podléhající principu historizace

Uchovává se u nich jak minulost, tak současnost. Tento typ obsahuje tabulky, ve kterých jsou uloženy informace o parcelách, budovách, jednotkách, OS, JPV, přiřazených listinách a katastrálních tělesech. Aktuálnost dat je vyjádřena pomocí následujících atributů: datum vzniku, datum zániku, stav dat a kontext změn. Může nastat několik kombinací atributů, jejich seznam a popis je uveden v tabulce č. 4.15.

Tabulka 4.15: Kombinace atributů vyjadřujících aktuálnost dat (zdroj: [23])

Operace	Stav dat	Kontext změn	Událost
UPDATE	-1	1	Objekt byl v exportovaném období změněn, původní verze objektu zanikla, nová verze vznikla. Nová verze není vzhledem k sys. datu aktuální, (záznam je v minulosti).
	-1	3	Objekt v exportovaném období vznikl a později byl změněn – verze není vzhledem k sys. datu aktuální, (záznam je v minulosti).
	0	3	Objekt byl v exportovaném období změněn, vznikla nová verze, která vzhledem k sys. datu je aktuální.
DELETE	3	1	Objekt byl v exportovaném období zrušen.
INSERT	0	3	Objekt vznikl v exportovaném období.
LOCK	0	2	Objekty zadané ve vstupních parametrech.

Jestliže je provedena operace UPDATE, tak je možné mít v exportovaném datovém souboru několik vět se stejným ID. Počet těchto vět je ovlivněn především počtem změn na daném objektu v exportovaném období, ale i vzájemným vztahem datových

položek *platnost od* a *platnost do* u exportovaného objektu vzhledem k danému období.

U operací INSERT a DELETE je v datovém souboru možný pouze jeden záznam s jedním ID.

Příklad obsahu změnového exportu:

```

1 ID N30;STAV_DAT N2;DATUM_VZNIKU D;DATUM_ZANIKU D;PRIZNAK_KONTEXTU
  N1;RIZENI_ID_VZNIKU N30;RIZENI_ID_ZANIKU N30
2 493589708;-1;"11.12.1998";"13.09.2002";1;908105708;919198708
3 493589708;-1;"13.09.2002";"14.11.2002";3;919198708;920435708
4 493589708;-1;"14.11.2002";"15.11.2002";3;920435708;920595708
5 493589708;-1;"15.11.2002";"";3;920595708;922200708
  
```

Tabulky nepodléhající principu historizace (skupina RIZENI)

Jsou exportovány daná řízení, která byla v zadaném intervalu (Platnost od – Platnost do) zplatněna, nebo uzavřena. Na tato řízení navazuje vazba na k.ú., objekty řízení, předměty říz., účast. říz., adresy, listiny, vazba na další údaje listin, typy účastníků řízení a denormalizovaná data o obeslání účastníků.

Tabulky nepodléhají principu historizace (skupiny GMPL a REZE)

U těchto tabulek se udržuje aktuální stav. Nemohou být proto obsahem změnových vět. Jsou zde udržovány informace o geometrických plánech (skupina GMPL) a rezervovaných číslech (skupina REZE).

Export číselníků

V číselnících jsou exportována jen ta data, u kterých byla platnost započata nebo ukončena v zadaném časovém intervalu (Platnost od – Platnost do).

Kapitola 5

VFK Plugin pro QGIS

Následující kapitola čerpá především ze zdrojů [4] a [12].

Prvotní verze zásuvného modulu QGIS **VFK Plugin** pro práci s daty českého katastru nemovitostí byla vyvinuta studenty oboru Geoinformatika na FSv ČVUT v Praze Annou Kratochvílovou a Václavem Petrášem.

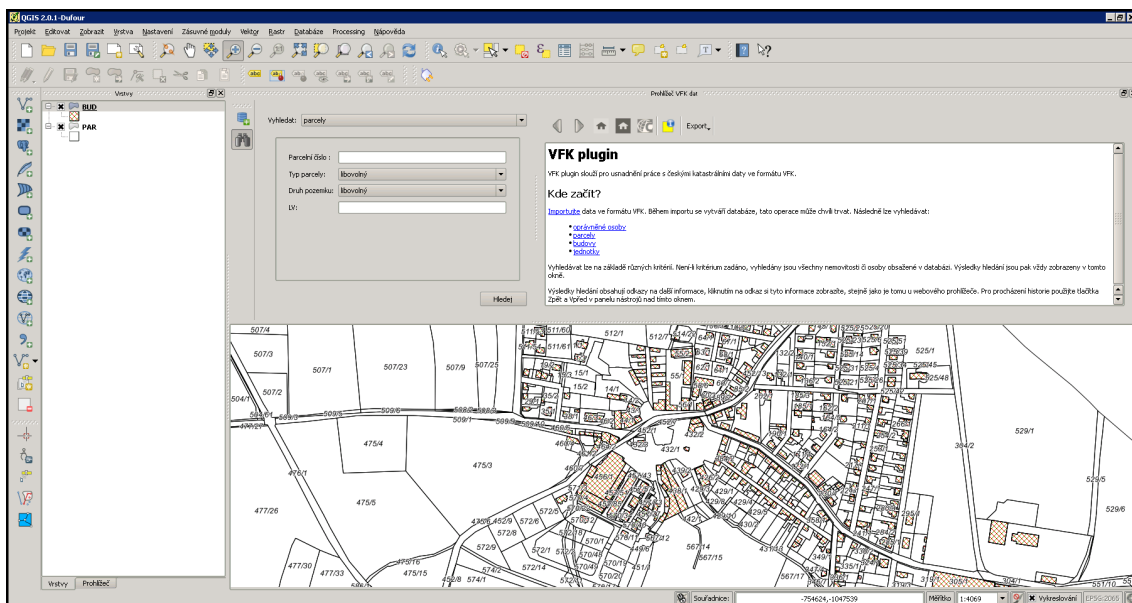
Zásuvný modul byl napsán v jazyce C++ s použitím knihovny Qt. Pracuje s daty v novém výměnném formátu katastru (VFK), viz kapitola č. 4.

Pro přístup k datům plugin využívá knihovny GDAL, respektive ovladače **VFK Driver** (viz 1.7). Načtená data ovladač ukládá do databáze SQLite, jejíž struktura je shodná se strukturou jednotlivých bloků v souboru VFK. Při opakovaném čtení dat ze stejného souboru je využívána již vytvořená SQLite databáze, oproti původnímu VFK souboru.

Zdrojové kódy zásuvného modulu jsou šířeny pod licencí GNU GPL¹. Zásuvný modul je možné stáhnout z oficiálního Git repositáře².

¹<https://raw.githubusercontent.com/ctu-osgeorel/qgis-vfk-plugin/master/LICENSE>

²<https://github.com/ctu-osgeorel/qgis-vfk-plugin-cpp>



Obrázek 5.1: VFK Plugin – C++ verze (zdroj: [4])

5.1 Funkcionalita

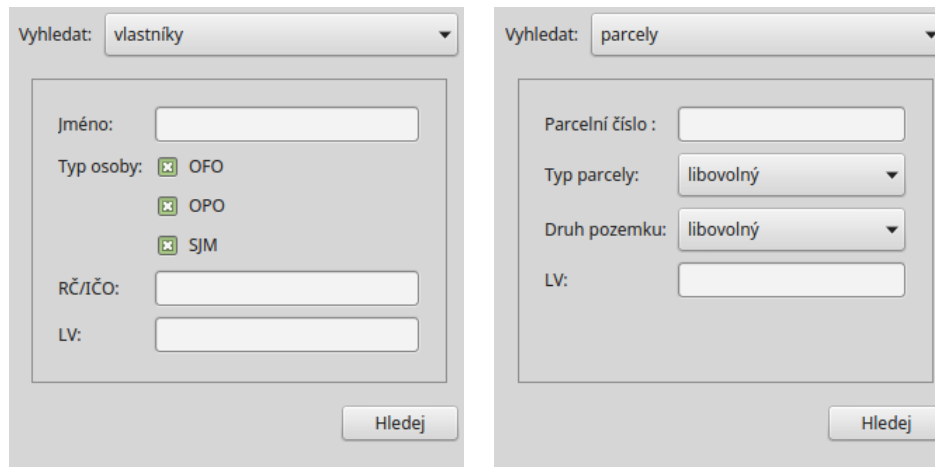
Cílem zásuvného modulu je zjednodušit práci s daty katastru nemovitostí. Plugin byl proto navržen tak, aby byl schopen jednoduše a rychle řešit základní úlohy nad těmito daty. Mezi základní funkcionalitu patří možnost vyhledávání podle:

- vlastníků,
- parcel,
- budov,
- jednotek.

Ve vedlejším okně, prohlížeči dat, se zobrazují výsledky vyhledávání. Tato data lze interaktivně procházet, podobně jako je tomu ve webovém prohlížeči, jelikož jsou ukládána ve formě HTML stránek. Jsou zde implementovány základní vlastnosti HTML. Pro navigaci mezi jednotlivými prvky jsou používány hypertextové odkazy. Obsah okna prohlížeče je ukládán do historie, což usnadňuje uživateli vyhledávání stále se opakujících informací (není potřeba opakovaně provádět stejný databázový dotaz). Historii lze procházet pomocí tlačítek „Vpřed“ a „Zpět“.

Pro zjištění aktuálního stavu o dané nemovitosti je zde implementováno propojení s aplikací **Nahlížení do katastru nemovitostí**³.

³<http://nahliznidokn.cuzk.cz>



(a) Vyhledávání vlastníků

(b) Vyhledávání parcel

Obrázek 5.2: Vyhledávací formuláře 1/2

Veškeré výpisy zobrazené v prohlížeči zásuvného modulu lze exportovat, a to do následujících formátů:

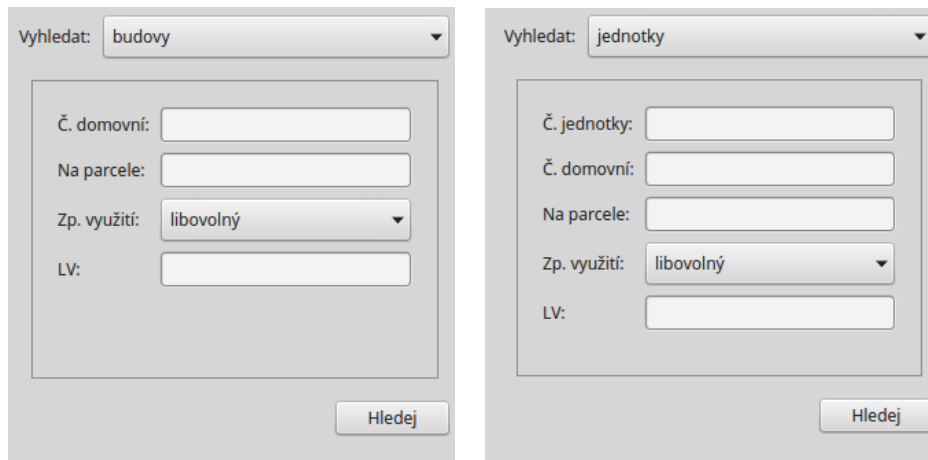
- **HTML:** umožňuje následné zobrazení ve webovém prohlížeči, případně import do textového procesoru,
- **zdrojový kód \LaTeX :** umožňuje vytvoření PDF nebo PS.

Zásuvný modul a mapové okno QGIS jsou navzájem propojené. Je tedy možné v mapovém okně zobrazit polohu vyhledaných prvků a opačně vyhledat informace o prvcích, které byly vybrány pomocí nástroje výběru QGIS právě v mapovém okně.

Uživatelská přívětivost používání zásuvného modulu je zvýšena „dokovatelností“ samotného okna pluginu. Modul je primárně ukotven k horní liště okna QGIS, ale je možné ho ukotvit i k ostatním okrajům. Okno modulu lze samozřejmě používat i samostatně (neukotvené).

Pro vrstvy parcel (PAR) a budov (BUD) byl vytvořen předdefinovaný vzhled ve formátu *XML*, respektive *QML*, se kterým QGIS nativně pracuje.

Součástí zásuvného modulu je také stručná nápověda, která je po spuštění pluginu zobrazena stejně jako veškeré hledané informace v prohlížeči dat ve formátu HTML. Nápověda může být vyvolána i po kliknutí na příslušné tlačítko v liště panelu nástrojů VFK Pluginu.



(a) Vyhledávání budov

(b) Vyhledávání jednotek

Obrázek 5.3: Vyhledávací formuláře 2/2

5.2 Použití pluginu

5.2.1 Instalace a spuštění

Stávající zásuvný modul, který je napsán v jazyce C++, je možné pod operačním systémem Ubuntu nainstalovat dle skriptů dostupných na stránkách pluginu⁴. Pro spuštění pod operačním systémem Windows, je potřeba stáhnout předkompilovanou verzi pro danou verzi QGIS a tu překopírovat do požadovaného adresáře. Podrobný návod je uveden opět na stránkách zásuvného modulu.

Takto nainstalovaný modul lze potom vyhledat v QGIS v nabídce Pluginy → Spravovat zásuvné moduly. Zde lze plugin dohledat po zadání filtru *VFK*. Spuštění pluginu je poté možné z hlavní nabídky QGIS Pluginy → VFK Plugin → VFK Plugin, nebo pomocí ikonky přímo z nástrojové lišty.

5.2.2 Rozložení

Okno zásuvného modulu se skládá ze tří hlavních částí. Vlevo se nachází hlavní panel nástrojů pro přepínání oken pro import souborů VFK a pro vyhledávání. Naprostou většinu pravé části tvoří prohlížeč dat, ve kterém se zobrazují výsledky vyhledávání a nápověda. Nad ním je poté nástrojová lišta, která obsahuje nástroje pro práci se zásuvným modulem.

⁴http://freegis.fsv.cvut.cz/gwiki/VFK/_/QGIS_plugin

5.2.3 Načítání dat

Pomocí zásuvného modulu je možné načíst soubor VFK. Tento import je k dispozici pomocí tlačítka „Procházet“ v nabídce *Import VFK*. V tomto okně je dále možné zvolit, které vrstvy (PAR, BUD) se budou zobrazovat v mapovém okně QGIS. V případě, kdy není zatrženo ani jedno pole, tak se načtou pouze popisná data. Bude tedy dostupné pouze vyhledávání bez interakce s mapovým oknem.

5.2.4 Vyhledávání

Po úspěšném načtení dat je zpřístupněno vyhledávání. Veškeré informace vyhledané o vlastnících, parcelách, budovách nebo jednotkách lze pohodlně procházet pomocí hypertextových odkazů v prohlížeči dat. Náhled vyhledaných informací je uveden na obrázku č. 5.4.

Informace o parcele

Parcelní číslo:	1291/1
List vlastnictví:	266
Výměra [m ²]:	2257
Určení výměry:	Ze souřadnic v S-JTSK
Typ parcely:	PKN
Mapový list:	DKM
Katastrální území:	Jáma 693936
Druh pozemku:	zahrada

[Sousední parcely](#)

Obrázek 5.4: Ukázka vyhledaných informací k dané parcele

Kapitola 6

Rozšíření stávající funkcionality

V této kapitole se budeme postupně věnovat jednotlivým krokům, které vedly k rozšíření stávajícího zásuvného modulu programu QGIS o podporu zpracování změnových vět souborů VFK. Spolu s tímto byly do pluginu přidány další funkcionality, jež budou zmíněny v dalším textu kapitoly.

6.1 Usnadnění distribuce pluginu

Jak již bylo zmíněno v kapitolách uvedených výše, stávající zásuvný modul byl napsán v jazyce C++. Program QGIS ale umožňuje ze vzdálených repositářů instalovat pouze moduly napsané v jazyce Python. C++ plugin tak musel být pro každou podporovanou platformu kompilován samostatně a instalován jiným vhodným způsobem.

Kompilace pod operačním systémem Linux (Ubuntu) může být provedena pomocí skriptů¹, které vytvořili autoři původní verze modulu.

Pod operačním systémem Microsoft Windows je situace poněkud komplikovanější. Pro každou verzi programu QGIS musí být plugin zkompilován samostatně vzhledem k typu operačního systému (32bit/64bit). Předkompilované verze mohou být staženy opět na stránkách pluginu, avšak nejsou zde uvedeny veškeré verze QGIS jak pro 32bit systém, tak pro 64bit systém. Uživatel jiné verzi QGIS (jiného typu operačního systému) by si tedy musel plugin zkompilovat sám, což může být i pro zkušeného uživatele nepřekonatelná překážka.

Výše zmíněné problémy značně snižují jak možnost většího rozšíření zásuvného modulu, tak komfort při jeho instalaci (aktualizaci). Řešením je přepis modulu do jazyka Python, čímž se plugin stane snadněji dostupným pro všechny uživatele bez rozdílu operačního systému, či verze programu QGIS.

¹Návod dostupný na: http://freegis.fsv.cvut.cz/gwiki/VFK/_/QGIS_plugin

6.1.1 Přepis do jazyka Python

Přepis zásuvného modulu do jazyka Python je důležitým bodem této práce vzhledem k usnadnění jeho distribuce. Při přepisu byl kód vytvářen co možná nejvíce podobný původnímu kódu v jazyku C++, a to hlavně z důvodu, aby byly zachovány veškeré funkcionality a vlastnosti tak, jak jsou známy stávajícím uživatelům zásuvného modulu.

Základní struktura kódu zásuvného modulu byla vygenerována pomocí doporučeného nástroje **Plugin Builder**² dostupného přímo z QGIS. Přepis byl značně ulehčen díky použití stejné knihovny Qt (respektive PyQt verze 4). V následujících ukázkách je porovnán původní kód v jazyce C++ (ukázka č. 6.1) s kódem v jazyce Python (ukázka č. 6.2). Oba kódy slouží pro přidání vektorové vrstvy ze souboru VFK do mapového okna QGIS.

Kód 6.1: Kód pro načtení vektorové vrstvy v C++

```

1 void VfkMainWindow::loadVfkLayer( QString vfkLayerName )
2 {
3     QgsDebugMsg( QString( "Loading vfk layer %1" ).arg( vfkLayerName ) );
4     if ( mLoadedLayers.contains( vfkLayerName ) )
5     {
6         QgsDebugMsg( QString( "Vfk layer %1 is already loaded" ).arg( vfkLayerName ) );
7         return;
8     }
9     QString composedURI = mLastVfkFile + "|layername=" + vfkLayerName;
10    QgsVectorLayer *layer = new QgsVectorLayer( composedURI, vfkLayerName, "ogr" );
11    mLoadedLayers.insert( vfkLayerName, layer->id() );
12    setSymbology( layer );
13
14    QList<QgsMapLayer *> myList;
15    myList << layer;
16    QgsMapLayerRegistry::instance()->addMapLayers( myList );
17 }

```

Kód 6.2: Kód pro načtení vektorové vrstvy v jazyce Python

```

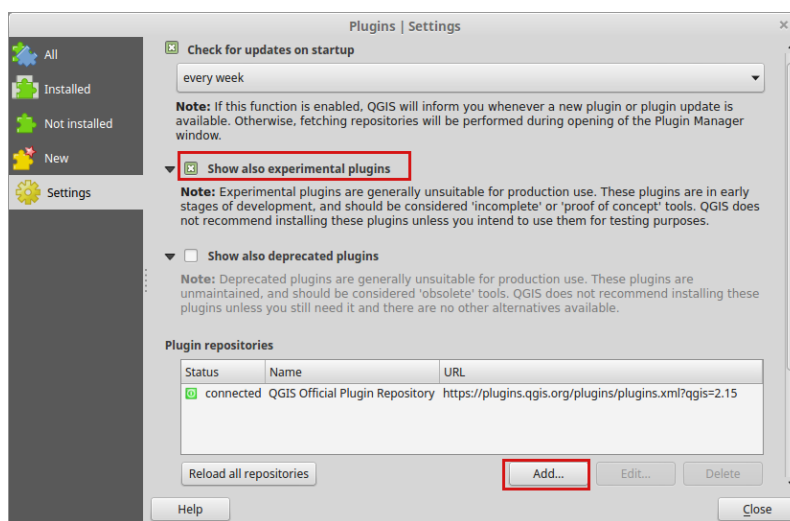
1 def __loadVfkLayer(self, vfkLayerName):
2     qDebug("\n(VFK) Loading vfk layer {}".format(vfkLayerName))
3     if vfkLayerName in self.__mLoadedLayers:
4         qDebug("\n(VFK) Vfk layer {} is already loaded".format(vfkLayerName))
5         return
6
7     composedURI = self.__mDataSourceName + "|layername=" + vfkLayerName
8     layer = QgsVectorLayer(composedURI, vfkLayerName, "ogr")
9     if not layer.isValid():
10        qDebug("\n(VFK) Layer failed to load!")
11
12    self.__mLoadedLayers[vfkLayerName] = layer.id()
13
14    try:
15        self.__setSymbology(layer)
16    except VFKWarning as e:
17        QMessageBox.information(self, 'Load Style', e, QMessageBox.Ok)
18
19    QgsMapLayerRegistry.instance().addMapLayer(layer)

```

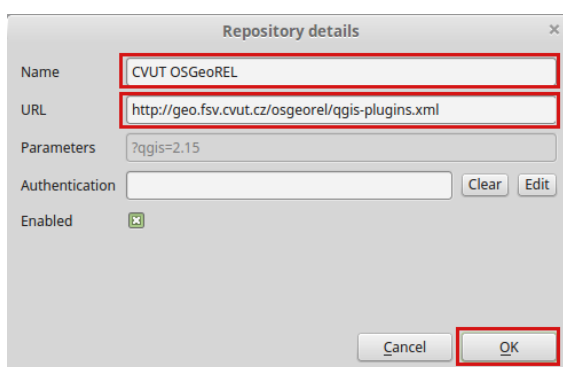
²<http://g-sherman.github.io/Qgis-Plugin-Builder/>

6.1.2 Instalace zásuvného modulu v QGIS

Díky přepisu zásuvného modulu do jazyka Python je plugin v QGIS dostupný standardním způsobem, podobně jako ostatní pluginy. Plugin v současné době není dostupný v oficiálním repositáři QGIS. Pro instalaci pluginu je potřeba do QGIS přidat nový repositář spravovaný organizací OSGeoREL³, pod kterou je tento plugin vyvíjen. Repositář je dostupný na adrese <http://geo.fsv.cvut.cz/osgeorel/qgis-plugins.xml>, viz obr. C.2. Modul je šířený jako experimentální, proto musí být tato volba zohledněna při jeho instalaci, viz obr. C.1. Okno pro zadání repositáře vyvoláme v záložce *Plugins* → *Manage and install plugins*.



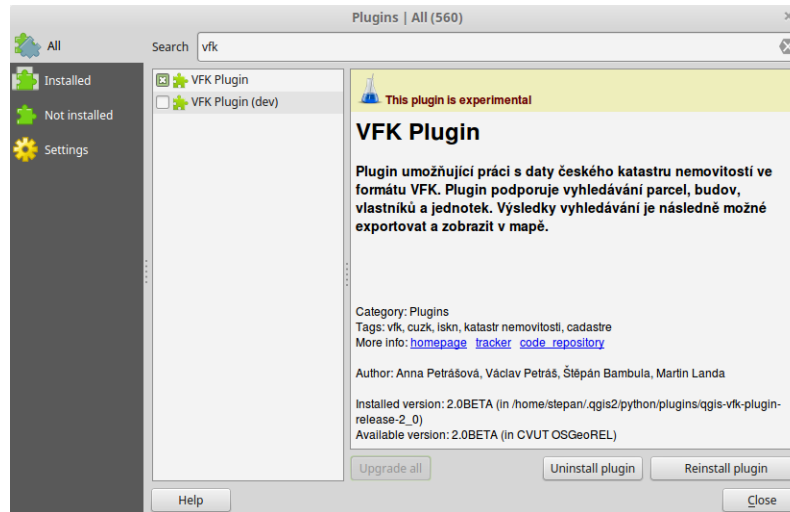
Obrázek 6.1: Přidání repositáře QGIS



Obrázek 6.2: Přidání repositáře QGIS – detail

Po přidání repositáře a jeho aktualizaci je zásuvný modul dohledatelný standardním způsobem pod názvem **VFK Plugin**, viz obr. C.3.

³<http://geomatics.fsv.cvut.cz/research/osgeorel>



Obrázek 6.3: Vyhledání zásuvného modulu

Po správné instalaci zásuvného modulu se do nástrojové lišty přidá jeho ikonka (viz obr. C.4). Plugin je poté dostupný po kliknutí na tuto ikonku nebo z menu *Plugins* → *VFK* → *Otevřít prohlížeč VFK*.



Obrázek 6.4: VFK Plugin – ikona

6.2 Úprava GDAL VFK driveru

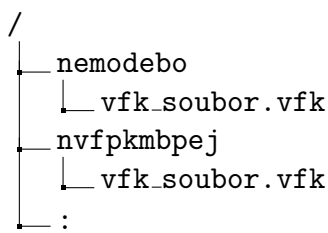
Před začátkem práce na podpoře zpracování změnových vět v QGIS VFK Pluginu bylo potřeba provést několik nezbytných úprav na straně GDAL VFK driveru, který plugin využívá pro čtení dat VFK, respektive jejich zápis do databáze SQLite.

Úprava byla provedena ve spolupráci s autorem GDAL VFK driveru Martinem Landou. Ten provedl většinu změn ovladače, které byly pro zpracování změnových VFK souborů nezbytné.

Veškeré provedené změny na ovladači VFK jsou dostupné v knihovně GDAL od verze 2.2.0. Přehled provedených změn popisuje „patch“, který je obsahem CD přiloženého k diplomové práci. Tento „patch“ je vytvořen proti vývojové verzi 2.2.0 knihovny GDAL.

6.2.1 Načítání dat z více souborů

Vzhledem ke struktuře, v jaké jsou změnová data dodávána od ČUZK uživateli, musela být implementována podpora načítání několika VFK souborů do jedné SQLite databáze. VFK soubory jsou k uživatelům distribuovány ve struktuře, která je uvedena v následující ukázce č. 6.5.



Obrázek 6.5: Struktura VFK souborů

K tomuto byla upravena funkce pro načítání dat. Původní chování VFK driveru bylo, že při prvním čtení VFK souboru se na pozadí ve stejném adresáři vytvořila databáze SQLite (její název byl shodný s názvem vstupního souboru), do které byla data načtena. Při opakovaném čtení VFK souboru se stejným názvem, se data četla již z existující databáze. Jméno výstupní SQLite databáze mohlo být ovlivněno pomocí proměnné prostředí `OGR_VFK_DB_NAME`, data poté byla ukládána/načítána do/z této databáze.

Nebylo tedy možné zpracovat data, která jsou uložena v adresářové struktuře (viz ukázka 6.5), kdy jsou některé VFK bloky uloženy v jednom adresáři a další ve druhém pod stejným názvem souboru.

Chování driveru bylo proto upraveno tak, že pokud uživatel potřebuje načíst více VFK souborů do jedné databáze, musí nejprve zadat cestu k výstupní SQLite databázi pomocí proměnné prostředí `OGR_VFK_DB_NAME`. Při čtení prvního souboru se načtou data standardním způsobem do databáze. Čtení dalších souborů je ošetřeno tak, že driver porovná tabulky v databázi s datovými bloky čteného VFK souboru. Jestliže tabulka daného jména v databázi neexistuje, tak ji vytvoří a pokračuje v načítání dat.

Při načítání dat driver testuje (viz 6.3), zda se jedná o změnový, nebo stavový VFK soubor. Jestliže se jedná o stavový, tak driver nepovolí načítání dat, která jsou již v databázi uložena (vznikaly by duplicity, což není u stavových dat možné). Při čtení změnového souboru je povoleno ukládání prvků s ID, které je již v tabulce uloženo (ve změnovém souboru je možné uložit prvky se stejným ID, viz kapitola 4.3). Kontrola uložení prvků s duplicitním ID je zajištěna pomocí databázového indexu `UNIQUE`, který je vytvářen právě jen pro stavová data.

Kód 6.3: Výňatek z kódu pro kontrolu změnového souboru

```

1  else if (pszLine[1] == 'H') {
2      /* check if amendment file */
3      if (EQUAL(pszLine, "&HZMENY;1")) {
4          m_bAmendment = TRUE;
5      }
6
7      /* header - metadata */
8      AddInfo(pszLine);
9  }

```

6.2.2 Čtení z databáze SQLite

Další podstatnou změnou na VFK driveru knihovny GDAL bylo přidání podpory pro čtení z databáze SQLite. Díky této nové funkcionalitě je možné v QGIS načíst vrstvy z databáze, do které byly aplikovány změny.

6.2.3 Vytváření geometrie prvků

Vytváření geometrie pro jednotlivé prvky, které jsou obsaženy ve VFK souboru, se děje pomocí metody `IVFKDataBlock::LoadGeometry()`. V dosavadní verzi VFK driveru byla tato metoda volána při požadavku o konkrétní prvek (například pomocí konzolového příkazu: `ogrinfo Export_vse.vfk PAR -FID 1`). Geometrie tedy nebyla sestavena dopředu po uložení dat do databáze, ale až poté, co si uživatel o daný prvek přímo zažádal.

Toto chování muselo být vzhledem ke struktuře, v jaké se k uživatelům dostávají VFK soubory, změněno. Geometrie se nyní sestavuje ihned po načtení dat do databáze. Jestliže databáze neobsahuje potřebné tabulky pro její sestavení, tak je zobrazeno upozornění⁴. Při čtení dalšího VFK souboru jsou data ukládána do stejné databáze a po jejich uložení se opět testuje, zda jsou k dispozici veškeré potřebné tabulky pro sestavení geometrie prvků. Jestliže ano, tak je geometrie sestavena. V opačném případě je opět zobrazeno upozornění.

⁴Upozornění, že v databázi chybí tabulka HP, potřebná pro sestavení geometrie parcel: „Warning 3: Data block HP not found. Unable to build geometry for PAR.“

6.3 Nově přidané funkcionality

Díky změnám provedeným ve VFK ovladači knihovny GDAL, bylo možné do zásuvného modulu QGIS přidat nové funkcionality nejen pro načítání dat.

Veškeré nové funkcionality související s načítáním více souborů, jsou do zásuvného modulu přidány pouze v případě, kdy je na počítači uživatele dostupná knihovna GDAL od verze 2.2.0. V případě nižší verze je upraven popisik ikonky pro přidání dalších VFK souborů (viz 6.4) a přidání dalších souborů není umožněno. Stejně tak je zakázáno načítání souborů z adresáře a databáze SQLite.

Kód 6.4: Kontrola verze GDAL na straně VFK pluginu

```

1  # check GDAL version
2  self.__gdal_version = int(gdal.VersionInfo())
3
4  if self.__gdal_version < 2020000:
5      self.actionZpracujZmeny.setEnabled(False)
6      self.pb_nextFile.setEnabled(False)
7      self.pb_nextFile.setToolTip(
8          u'Neni mozne nacist vice souboru, verze GDAL je nizsi nez 2.2.0.')
9      self.actionZpracujZmeny.setToolTip(u'Zpracovani zmen neni povoleno, verze GDAL je nizsi nez
10     2.2.0.')
11     self.groupBox.setEnabled(False)

```

6.3.1 Čtení z databáze SQLite

Zásuvný modul umožňuje načítání souboru ve formátu databáze SQLite. Tato možnost uživateli umožní práci s již zpracovanými VFK daty pomocí ovladače VFK knihovny GDAL. Tato volba je užitečná zejména v případě, kdy uživatel potřebuje načíst objemná data, případně při načítání databáze s aplikovanými změnami (viz kapitola 6.4).

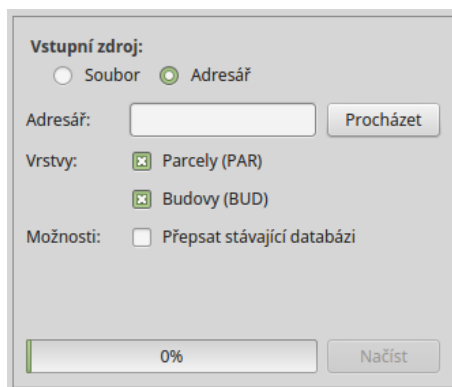
6.3.2 Načítání více souborů

Načítání více souborů je povoleno, jestliže uživatel zvolí jako vstupní zdroj (viz obr. 6.7 prvek č. 1) volbu „Soubor“. Poté je uživateli zpřístupněna ikona, díky níž je možné přidávat cestu k více VFK souborům. Počet takto načtených VFK souborů je omezen na pět. Není povoleno načítání více databází SQLite.

6.3.3 Načítání VFK souborů z adresáře

Jestliže má uživatel VFK data uložena ve více souborech a nemůže je načíst jednotlivě, je vhodné použít tuto volbu. Tato volba je zpřístupněna po změně vstupního zdroje z hodnoty „Soubor“ na hodnotu „Adresář“. Následně je změněn vzhled „widgetu“ pro načítání, viz obr. 6.6.

Při této volbě je umožněn pouze jeden vstup. Tím je cesta k adresáři, ze kterého bude načítání probíhat. Zásuvný modul poté stromově projde obsah zadaného adresáře a vyhledá všechny VFK soubory, které budou načteny.



Obrázek 6.6: VFK Plugin – okno pro načítání z adresáře

6.3.4 Automatické pojmenování databáze

Jak bylo řečeno v kapitole 6.2, ovladač VFK knihovny GDAL načítá VFK soubory interně do databáze SQLite. Tato databáze je umístěna o adresář výše, než první načítaný soubor. Její jméno je tvořeno jako `jmeno_prvniho_souboru.XXX.db`, kde `XXX` nabývá hodnot „stav“ pro stavová data, nebo „zmeny“ pro data obsahující změnové věty.

6.3.5 Přepsání existující databáze

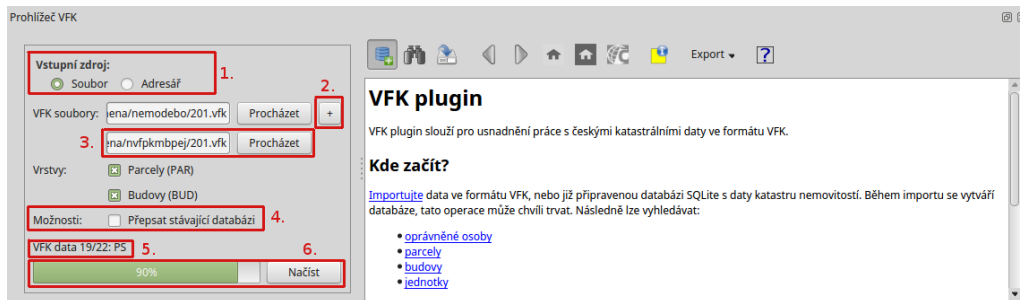
Tato volba umožňuje přepsat stávající databázi SQLite před načtením dat VFK. Přepsání současné databáze je užitečné, pokud uživatel nechce pracovat se stávající databází. Jedním z důvodů pro tento krok může být nekompatibilita dat načtených GDAL ve verzi 1.11 s novou verzí knihovny GDAL (2.0). Přepsání databáze je zajištěno pomocí vyexportování proměnné prostředí `OGR_VFK_DB_OVERWRITE=1`, která náleží k ovladači VFK v knihovně GDAL.

6.3.6 Změna vzhledu zásuvného modulu

Jak z výše uvedeného vyplývá, k přidání nových funkcionalit musel být změněn i vzhled samotného okna zásuvného modulu pro načítání dat. Nový vzhled je uveden na obrázku č. 6.7, kde byly přidány následující prvky:

1. Volba vstupního zdroje,

2. tlačítko pro vložení nového řádku do formuláře pro přidání souboru,
3. nově vygenerovaný formulář pro vyhledání VFK souboru,
4. možnost přepsání již existující databáze SQLite,
5. popisek stavu načítání dat,
6. grafický indikátor stavu načítání.



Obrázek 6.7: VFK Plugin – okno pro načítání souborů

6.4 Zpracování změnových vět

Schopnost zpracovat změnové věty VFK souborů je důležitá pro každého uživatele, který s daty katastru nemovitostí pracuje. Jestliže uživatel již jednou vlastní stavová data pro dané území a potřebuje hlídat aktuálnost těchto dat, je pro něj výhodnější stahovat právě změnové VFK soubory, než soubory stavové.

Zpracování těchto vět pomocí zásuvného modulu v QGIS nebylo do této doby možné. O implementaci podpory pro jejich zpracování na straně VFK pluginu pojednává právě tato kapitola.

Změnové věty jsou směrem k uživatelům šířeny ve stejné struktuře jako stavová data, viz obr. 6.5. Pro jejich zpracování musely být nejprve provedeny úpravy na straně VFK driveru v knihovně GDAL, viz kapitola 6.2. Díky tomuto kroku a úpravě principu načítání souborů v okně zásuvného modulu (viz obr. 6.7) je možné změnové VFK soubory uložit do jedné databáze a pomocí VFK pluginu zobrazit v programu QGIS. V případě, kdy změny obsahují potřebné datové bloky, je nad změnovými daty umožněno i vyhledávání.

6.4.1 Metodika aplikace změn na stavová data

Aplikace změn na stavová data předpokládá, že uživatel již má pomocí VFK driveru knihovny GDAL připravenou databázi jak se stavovými daty, tak s daty

změnovými. V opačném případě může uživatel tyto databáze vytvořit pomocí samotného VFK pluginu.

Pro aplikaci změn byla vytvořena samostatná třída (`ApplyChanges`), pomocí které mohou být změny aplikovány. Toto může být provedeno jednak pomocí zásuvného modulu (viz 6.4.2), nebo pomocí samostatně spustitelné konzolové aplikace (viz 6.5).

Vstupními argumenty pro tuto třídu, respektive metodu `run(...)`, jsou:

- **Hlavní databáze:** Databáze, která obsahuje stavová data, na něž budou změny aplikovány.
- **Změnová databáze:** Databáze obsahující změnová data.
- **Režim „debug“:** Nepovinný argument. V případě jeho zadání jsou na standardní výstup vypisovány veškeré SQL příkazy do databáze. Implicitně je nastaven na hodnotu `False`.

Jediným výstupním argumentem je poté cesta k nově vytvořené databázi, která bude obsahovat stavová data s aplikovanými změnami.

Kód 6.5: Nápopěda ke konzolové aplikaci pro implementaci změn

```

1 $ ./applyChanges.py -h
2
3 Usage: applyChanges.py [-h] [-v] -i INPUT -c CHANGES -o OUTPUT [-d
4 ]
5
6 Script applies changes from amendment VFK database to main VFK
7 database. In this process new database is created.
8
9 optional arguments:
10  -h, --help          Show this help message and exit
11  -v, --version       Show program's version number and exit
12  -i INPUT, --input INPUT
13                      Path to the main database.
14  -c CHANGES, --changes CHANGES
15                      Path to the database with changes.
16  -o OUTPUT, --output OUTPUT
17                      Path to the new database which will be
18                      created.
19  -d, --debug         Enables debug mode.
```

V prvním kroku se pomocí nástroje `shutil.copy2(src, dst)` vytvoří kopie vstupní databáze. Cesta k ní je zadána pomocí výstupního argumentu. Čas potřebný pro tvorbu kopie databáze je přímo úměrný její velikosti. Tento krok proto může být

pro objemná data časově náročný. Díky tvorbě kopie a práce s ní je ale zajištěno, že v případě, kdyby se aplikace změn nepovedla, uživatel nepřijde o původní data.

V dalším kroku se příkazem `sqlite3.connect(database)` provede připojení ke kopii databáze (v dalším textu bude tato kopie označována jako výstupní / „output“). Databáze se změnami je poté k této výstupní databázi připojena pomocí SQL příkazu `ATTACH DATABASE`⁵. Práce s takto připojenou databází je poté shodná jako se schématy, známými například z PostgreSQL.

Samotné změny jsou aplikovány pouze z tabulek, které podporují princip „historizace“ (viz kapitola 4.3) a které nejsou prázdné. Tyto tabulky musí obsahovat sloupec `STAV_DAT`. Pro výběr odpovídajících tabulek slouží k tomu určená metoda `_findTablesWithChanges()`.

Při aplikaci změn jsou v cyklu pro každou tabulku spouštěny databázové příkazy `DELETE` (viz ukázka 6.6) a `INSERT` (viz ukázka 6.7), kde proměnná `table` označuje jméno databázové tabulky, proměnná `column` jméno příslušného sloupce, `columns` označuje seznam sloupců a proměnná `id` je nahrazena identifikátorem vkládaného prvku z databáze, která obsahuje změnová data.

Příkaz `DELETE` z výstupní databáze smaže veškeré prvky, které se vyskytují v databázi se změnovými daty. Tento příkaz je vykonán, protože v databázi se změnovými daty jsou vždy aktuálnější informace o prvcích, než v databázi původní.

Kód 6.6: Aplikace změn – příkaz DELETE

```

1 DELETE FROM main.{table}
2 WHERE id IN (
3 SELECT DISTINCT t1.id FROM main.{table} t1
4 JOIN db2.{table} t2 ON t1.id = t2.id);

```

Kód 6.7: Aplikace změn – příkaz INSERT

```

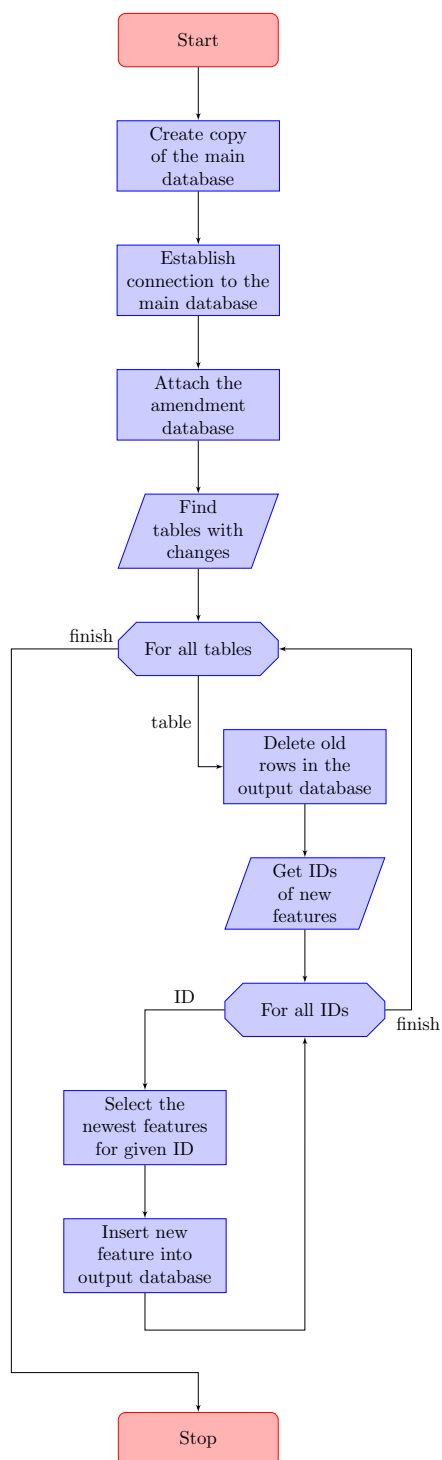
1 INSERT INTO main.{table}
2 SELECT {columns} FROM db2.{table}
3 WHERE ogr_fid={selected_ogr_fid}
4 AND stav_dat=0
5 AND id = {id};

```

Po smazání veškerých neaktuálních záznamů z výstupní databáze, je vykonán příkaz `INSERT`. Tento příkaz je spuštěn v cyklu nad všemi záznamy ze změnové databáze, které jsou určeny pro import. V případě, kdy jsou ve změnové databázi prvky, na kterých bylo v exportovaném období použito více operací (`UPDATE`, `INSERT`, `DELETE`), jsou příslušné prvky seřazeny podle data vzniku a do databáze je vkládán pouze nejnovější prvek (ten je jednoznačně vybrán dle identifikátoru `selected_ogr_fid`).

Pro lepší orientaci o průběhu aplikování změn na stavová data byl vytvořen diagram, viz obr. 6.8.

⁵https://www.sqlite.org/lang_attach.html



Obrázek 6.8: Diagram průběhu implementace změn na stavová data

6.4.2 Implementace na straně zásuvného modulu

Funkcionalita pro aplikování změn z prostředí zásuvného modulu byla implementována díky vytvoření nového formuláře v okně pluginu. Na okně zásuvného modulu byly provedeny následující úpravy:

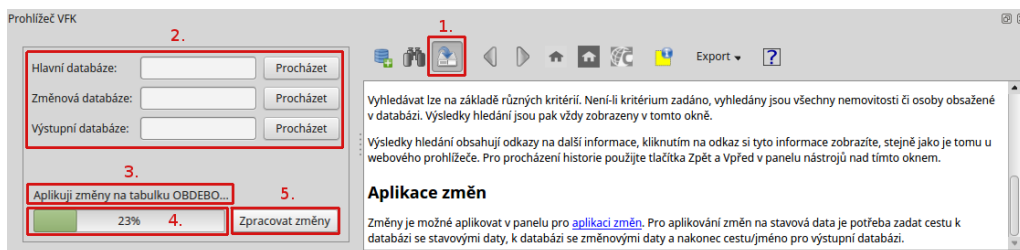
1. **Ikona změn:** Okno pro aplikaci změn na stavová data je vyvoláno pomocí nově vytvořené ikony v nástrojové liště, viz obr. C.8.



Obrázek 6.9: VFK Plugin – ikona pro zpracování změn (zdroj: <https://goo.gl/BiHWLz>)

2. **Formuláře:** K dispozici jsou tři formuláře pro zadání cesty ke vstupní databázi se stavovými daty, k databázi se změnovými daty a databázi výstupní, do které budou uložena stavová data s aplikovanými změnami.
3. **Popisek průběhu zpracování:** V tomto popisku jsou zobrazovány informace o průběhu zpracování změn.
4. **Grafický indikátor průběhu:** Zde je průběh zpracování změn vyjádřen procentuálně a graficky.
5. **Tlačítko pro zpracování:** Po stisku tohoto tlačítka je vyvolána metoda `run(...)`, díky které jsou změny aplikovány do databáze se stavovými daty.

Funkcionalita pro aplikaci změn obsažených v databázi SQLite na data obsažená v databázi se stavovými daty je do zásuvného modulu přidána pouze v případě, kdy je na počítači uživatele dostupná knihovna GDAL ve verzi 2.2.0 a vyšší. V opačném případě je ikonka pro zpracování změn neaktivní.



Obrázek 6.10: VFK Plugin – okno pro zpracování změn

Závěr

Cílem diplomové práce bylo rozšířit stávající zásuvný modul pro práci s katastrálními daty v programu QGIS. Samotnému rozšíření modulu o nové funkcionality předcházelo jeho přepsání do jazyka Python, díky čemuž se velice usnadnila jeho distribuce směrem k uživatelům. Hlavním tématem diplomové práce bylo zpracování změnových vět výměnného formátu katastru nemovitostí.

Práce byla rozdělena do dvou částí – teoretické a praktické. V teoretické části jsem se věnoval rešerši dostupných nástrojů pro práci s katastrálními daty a použité technologii, která byla nezbytná pro vypracování práce. Na tyto kapitoly navazuje část týkající se Informačního systému katastru nemovitostí, ze kterého jsou datové soubory VFK exportovány. Hlavním bodem teoretické části je kapitola týkající se výměnného formátu katastru nemovitostí, kde jsem kladl důraz především na jeho strukturu. Pochopení struktury výměnného formátu bylo nezbytné pro správné navržení zásuvného modulu. V této kapitole jsem se zaměřil i na strukturu změnových vět VFK tak, aby mohl být vypracován hlavní bod diplomové práce – **zpracování změnových vět**.

Na teoretickou část volně navazuje kapitola týkající se stavu současného zásuvného modulu pro zpracování katastrálních dat v QGISu. Zde jsem se snažil popsat práci s modulem od jeho instalace a zprovoznění, přes načítání souborů VFK, až k jeho klíčovým funkcionalitám týkajících se samotného vyhledávání nad importovanými katastrálními daty. Nechybí ani informace o možnostech exportu výsledků vyhledávání do některého z podporovaných formátů.

Ve stěžejní kapitole praktické části – Rozšíření stávající funkcionality (6) – jsem popsal veškeré změny provedené nejen na straně zásuvného modulu. V kapitole jsou zmíněny kroky, které vedly ke zlepšení dostupnosti modulu, změny na straně ovladače VFK v knihovně GDAL, nově přidaná funkcionality týkající se zpracování změnových vět a ostatní přidané vylepšení zásuvného modulu.

Přepis stávajícího zásuvného modulu do jazyka Python byl započat v rámci semestrálního projektu předmětu *Free software GIS* v roce 2015. Na přepisu prvních částí kódu (zejména pro načítání VFK souboru) se podíleli i ostatní členové skupiny pracující na tomto tématu, viz stránky předmětu⁶. Kód v jazyce Python se snaží

⁶http://geo.fsv.cvut.cz/gwiki/155YFSG_Free_software_GIS

co možná nejvíce kopírovat strukturu původního kódu v jazyce C++. Modul byl navržen s pomocí stejné knihovny Qt, respektive PyQt, která je jednou z hlavních knihoven pro tvorbu uživatelského rozhraní a značně ulehčuje tvorbu celé aplikace. Zásuvný modul komunikuje s prostředím QGIS díky použití aplikačního rozhraní QGIS API.

Poté, co byl stávající zásuvný modul přepsán, tak byl s pomocí vedoucího diplomové práce umístěn do nově vzniklého repositáře⁷, který spravuje organizace OSGeo-REL na ČVUT v Praze. Přepsaný plugin je v repositáři dostupný ve verzi 2.0BETA a jeho funkcionality kopíruje původní zásuvný modul v C++. Pro tuto verzi modulu byla v oficiálním repositáři se zdrojovým kódem⁸ vytvořena nová vývojová větev `release-2.0`. Modul je v této formě dostupný pro uživatele po přidání výše zmíněného repositáře do programu QGIS, viz C.1.

Funkcionality týkající se zpracování změnových vět předcházela úprava ovladače VFK v knihovně GDAL. Změny na ovladači byly provedeny spolu s autorem Martinem Landou. Jako jedna z nejdůležitějších může být zmíněna podpora načítání více souborů do jedné databáze SQLite, či úprava struktury výstupní databáze podle toho, zda je načten stavový, nebo změnový soubor VFK. Další důležitou novou vlastností ovladače je možnost čtení přímo z databáze SQLite, což dovoluje do QGIS načíst databázi, na kterou byly aplikovány změnové věty.

Pro zpracování změnových vět vznikla samostatná třída, která umožňuje aplikovat změny na stavová data. Jak změnová, tak stavová data musí být uložena v předem připravených databázích SQLite. Výstupem je nově vzniklá databáze s již obsaženými změnami. Změny mohou být aplikovány jednak pomocí vzniklé konzolové aplikace, nebo přímo v zásuvném modulu.

Nově vzniklé funkcionality jsou dostupné od verze 2.1BETA zásuvného modulu a jsou k dispozici pro všechny uživatele, kteří mají nainstalovanou knihovnu GDAL ve verzi 2.2.0 a vyšší. Vývoj této verze probíhal (a stále probíhá) ve větvi `master` opět na serveru Github, kde jsou k dispozici veškeré zdrojové kódy. Mezi nejpodstatnější nově vzniklé funkcionality, které jsou v této verzi modulu obsaženy, patří podpora načítání a zpracování změnových souborů, případně možnost načtení více souborů VFK do jedné databáze, nebo čtení z databáze SQLite. V zásuvném modulu nejen této verze bylo opraveno i několik chyb, obsažených v původní verzi pluginu.

Podněty pro nové funkcionality zásuvného modulu pocházely přímo od uživatelů předešlé C++ verze. Vývoj třídy pro aplikaci změn, respektive metodika samotného nahrazování stavových dat změnovými, byl také konzultován s lidmi z praxe, kteří mají se zpracováním změn bohaté zkušenosti.

Další vývoj zásuvného modulu se může ubírat mnoha směry. Jednou z možných budoucích funkcionalit může být podpora exportu do více formátů – především tedy do PDF, či XML. Vhodné by také bylo, kdyby modul lépe pracoval s více vlákny

⁷<http://geo.fsv.cvut.cz/osgeorel/qgis-plugins.xml>

⁸https://github.com/ctu-osgeorel/qgis-vfk-plugin/tree/release-2_0

při načítání VFK souborů, toto se týká hlavně fáze při načítání datového souboru. Dalším bodem může být zobrazení sluček v mapovém okně, nebo hromadný export podrobností o nemovitostech. Obě tyto funkcionality vzešly na základě požadavků od uživatelů. Jako nejdůležitější bod ale vnímám překlad zásuvného modulu do angličtiny, což v současné chvíli blokuje přesun modulu do hlavního repositáře QGIS. Tímto by se modul stal pro uživatele ještě více dostupnější než doposud.

Hlavním výstupem diplomové práce je tedy zásuvný modul pro práci s katastrálními daty v prostředí QGIS. Tento modul disponuje několika novými funkcionalitami, kde nejdůležitější z nich je podpora zpracování změn. Zdrojové kódy jsou dostupné v repositáři na serveru Github (<https://github.com/ctu-osgeorel/qgis-vfk-plugin>). Modul může být v prostředí QGIS nainstalován po přidání nového repositáře (viz poznámka č. 7 výše), který je spravovaný organizací OSGeoREL. Spolu se zásuvným modulem vznikla i uživatelská příručka, která je v diplomové práci uvedena jako příloha C.

Závěrem bych chtěl upozornit, že se jedná o vývojovou verzi zásuvného modulu a budu rád za veškeré podněty pro jeho další vývoj, ideálně ve formě tzv. „Issues“ na stránce se zdrojovým kódem.

Literatura

- [1] AbcLinuxu - výkladový slovník. [online], cit. 2016-03-15.
URL <http://www.abclinuxu.cz/slovník/open-source>
- [2] CAD Studio – VFK2DB. [online], cit. 2016-04-18.
URL <http://www.cadstudio.cz/vfk>
- [3] CAD Studio – VFK2DWG. [online], cit. 2016-04-18.
URL <http://www.cadstudio.cz/vfk2dwg>
- [4] VFK / QGIS plugin. [online], cit. 2016-04-26.
URL http://freegis.fsv.cvut.cz/gwiki/VFK/_QGIS_plugin
- [5] GDAL - Official website. [online], cit. 2016-03-15.
URL <http://www.gdal.org/>
- [6] GDAL - Wikipedie. [online], cit. 2016-03-16.
URL <https://cs.wikipedia.org/wiki/GDAL>
- [7] VFK – Czech Cadastral Exchange Data Format. [online], cit. 2016-04-21.
URL http://gdal.org/drv_vfk.html
- [8] Import dat KN ve výměnném formátu. [online], cit. 2016-04-21.
URL <http://www.gisoft.cz/Moduly/ImportVFK>
- [9] Informační systém katastru nemovitostí. [online], cit. 2016-03-28.
URL <http://goo.gl/9o771D>
- [10] ISKN pro ArcGIS for Desktop. [online], cit. 2016-04-20.
URL <https://www.arcdata.cz/produkty/ceska-specifika/iskn-pro-arcgis-for-desktop>
- [11] JEDLIČKA, K.; JEŽEK, J.; PETRÁK, J.: Otevřený katastr - svobodné internetové řešení pro prohlížení dat výměnného formátu katastru nemovitostí. *Geoinformatics FCE CTU*, , č. 2, 2007: s. 111–118, ISSN 1802-2669 [online].
- [12] KRATOCHVÍLOVÁ, A.; PETRÁŠ, V.: Quantum GIS plugin for Czech cadastral data. *Geoinformatics FCE CTU*, , č. 8, 2012: s. 91–98, ISSN 1802-2669 [online].

- [13] LANDA, M.: OGR VFK Driver Implementation Issues. *GIS Ostrava 2010*.
URL <http://geo.fsv.cvut.cz/~landa/publications/2010/gis-ostrava-2010/paper/landa-ogr-vfk.pdf>
- [14] LANDA, M.: *Návrh modulu GRASSu pro import dat ve výměnném formátu ISKN*. Diplomová práce, České vysoké učení technické, Praha, 2005.
URL <http://goo.gl/5ZzQps>
- [15] MCINERNEY, D.; KEMPENEERS, P.: *Open Source Geospatial Tools: Applications in Earth Observation (Earth Systems Data and Models), Earth Systems Data and Models (Book 3)*, ročník 4. Springer, 2015 vydání, 11 2014, ISBN 978-3319018232.
- [16] PILGRIM, M.: *Dive into Python*. Berkeley: Apress, 7 2004, ISBN 978-1-59059-356-1.
- [17] PyQt - Official website. [online], cit. 2016-03-15.
URL <https://riverbankcomputing.com/software/pyqt/intro>
- [18] PyQt - Wikipedie. [online], cit. 2016-03-15.
URL <https://en.wikipedia.org/wiki/PyQt>
- [19] Python - official website. [online], cit. 2016-03-15.
URL <https://www.python.org/>
- [20] QGIS - Official website. [online], cit. 2016-03-15.
URL <http://www.qgis.org/>
- [21] QGIS - Wikipedie. [online], cit. 2016-03-16.
URL <https://cs.wikipedia.org/wiki/QGIS>
- [22] Spirit VFK. [online], cit. 2016-04-21.
URL <http://www.georeal.cz/cz/spirit-desktop/spirit-vfk>
- [23] Struktura VFK. [online], cit. 2016-03-15.
URL <http://goo.gl/2I22N6>
- [24] Summerfield, M.: *Rapid GUI programming with Python and Qt: the definitive guide to PyQt programming*. Prentice Hall: Upper Saddle River, 2008, ISBN 978-0132354189.
- [25] Topol VFK Import. [online], cit. 2016-04-20.
URL <http://www.datasystem.cz/vfk-import-s-37-m-4.html>
- [26] Slovník VUGTK. [online], cit. 2016-04-01.
URL <http://www.vugtk.cz/slovník/index.php>

- [27] Výměnný formát ISKN v textovém tvaru. [online], cit. 2016-03-30.
URL <http://goo.gl/zF0Dt2>
- [28] Výměnný formát KN před ISKN. [online], cit. 2016-03-30.
URL <http://goo.gl/XRjxyb>

Seznam obrázků

1.1	Aplikace ISKN Studio	4
1.2	Aplikace ISKN View	4
1.3	Import dat KN ve výměnném formátu – ukázka použití	5
1.4	Spirit VFK – ukázka aplikace	6
1.5	Aplikace VFK2DWG	7
1.6	Topol VFK Import – ukázka zpracovaných dat	8
1.7	Ukázka načtení VFK pomocí VFK Driveru GDAL v programu GRASS GIS	10
2.1	QGIS – logo	11
2.2	GDAL – logo	12
2.3	Python – logo	12
2.4	PyQt – logo	13
3.1	ČUZK – logo	14
5.1	VFK Plugin – C++ verze	33
5.2	Vyhledávací formuláře 1/2	34
5.3	Vyhledávací formuláře 2/2	35
5.4	Ukázka vyhledaných informací k dané parcele	36
6.1	Přidání repositáře QGIS	39
6.2	Přidání repositáře QGIS – detail	39
6.3	Vyhledání zásuvného modulu	40
6.4	VFK Plugin – ikona	40
6.5	Struktura VFK souborů	41

6.6	VFK Plugin – okno pro načítání z adresáře	44
6.7	VFK Plugin – okno pro načítání souborů	45
6.8	Diagram průběhu implementace změn na stavová data	48
6.9	VFK Plugin – ikona pro zpracování změn	49
6.10	VFK Plugin – okno pro zpracování změn	49
C.1	Přidání repositáře QGIS	VIII
C.2	Přidání repositáře QGIS – detail	IX
C.3	Vyhledání zásuvného modulu	IX
C.4	VFK Plugin – ikona	IX
C.5	VFK Plugin – rozložení prvků	X
C.6	Vyhledávací formuláře 1/2	XII
C.7	Vyhledávací formuláře 2/2	XII
C.8	VFK Plugin – ikona pro zpracování změn	XIII
C.9	VFK Plugin – okno pro zpracování změn	XIII

Seznam tabulek

4.1	Datové skupiny VF ISKN	20
4.2	Seznam položek hlavičky	21
4.3	Ukázka hlavičky	21
4.4	Datové typy ISKN	23
4.5	Seznam datových bloků ve skupině „NEMOVITOSTI“	24
4.6	Seznam datových bloků ve skupině „JEDNOTKY“	25
4.7	Seznam datových bloků ve skupině „VLASTNICTVÍ“	25
4.8	Seznam datových bloků ve skupině „JINÉ PRÁVNÍ VZTAHY“	26
4.9	Seznam datových bloků ve skupině „ŘÍZENÍ“	26
4.10	Seznam datových bloků ve skupině „PRVKY KATASTRÁLNÍ MAPY“	27
4.11	Seznam datových bloků ve skupině „BPEJ“	27
4.12	Seznam datových bloků ve skupině „GEOMETRICKÝ PLÁN“	28
4.13	Seznam datových bloků ve skupině „REZERVOVANÁ ČÍSLA“	29
4.14	Seznam datových bloků ve skupině „ADRESNÍ MÍSTA“	29
4.15	Kombinace atributů vyjadřujících aktuálnost dat	30

Seznam ukázek kódu

6.1	Kód pro načtení vektorové vrstvy v C++	38
6.2	Kód pro načtení vektorové vrstvy v jazyce Python	38
6.3	Výňatek z kódu pro kontrolu změnového souboru	42
6.4	Kontrola verze GDAL na straně VFK pluginu	43
6.5	Nápověda ke konzolové aplikaci pro implementaci změn	46
6.6	Aplikace změn – příkaz DELETE	47
6.7	Aplikace změn – příkaz INSERT	47

Příloha A

Popis tříd zásuvného modulu

ApplyChanges

Třída slouží pro aplikaci změn na stavová data. Pracuje s databázemi SQLite, které jsou vytvořeny pomocí VFK Driveru GDAL. Podrobnější popis zpracování změnových souborů touto třídou je uveden v kapitole 6.4.

BudovySearchForm

Jedná se o třídu, která se stará o vyhledávací prvky ve formuláři pro vyhledávání budov. Vzhled uživatelského rozhraní této třídy byl navržen samostatně pomocí programu Qt Designer. Kód návrhu je uložen v souboru `ui_budovysearchform.ui`, respektive v překompilované verzi pro jazyk Python `ui_budovysearchform.py`. Ukázka konstruktoru třídy je uvedena v příloze B.1.

DocumentBuilder

V třídě `DocumentBuilder` sestavuje dokument, který bude zobrazen v prohlížeči dat (nebo exportován do některého z podporovaných formátů), na základě výsledků vyhledávání. O správné sestavení se stará metoda `buildHtml()`, viz příloha B.2.

Třída mimo jiné obsahuje několik metod pro správné sestavení tabulek (např. `tableParcely()`), metody pro tvorbu jednotlivých částí listu vlastnictví jako je `partTelesoB1()`, nebo metody pro zobrazení detailu o vyhledávaném tělese (např. `pageBudova()`).

Důležitou metodou v této třídě je metoda `pageHelp()`, díky které je možné zobrazit krátkou nápovědu k zásuvnému modulu, a metoda `saveDefinitionPoint()`, která slouží k získání a uložení definičního bodu vyhledané nemovitosti.

Domains

V této třídě jsou jednoduché metody pro zjištění domény prvku na základě vstupního argumentu. Je zde metoda `anoNe()`, která testuje, zda je vstupní argument rovný „a“, a podle toho vrací `True/False`. Metoda `cpCe()` převádí hodnotu vstupního argumentu na řetězec „Číslo popisné“ / „Číslo evidenční“. Metoda `druhUcastnika()` kontroluje, o jaký typ subjektu se jedná (např. právnická osoba). Poslední metodou je metoda `rodinnyStav()`, která vrací rodinný stav subjektu na základě zadaného vstupního kódu.

HtmlDocument

Třída `HtmlDocument` je potomkem třídy `VfkDocument`. Jsou v ní obsaženy metody pro tvorbu jednotlivých částí kódu v jazyce *HTML*. Vhodnou kombinací těchto částí je poté sestaven celý *HTML* dokument. Takto připravený dokument je poté možné vyexportovat pomocí příslušné funkcionality a zobrazit v libovolném prohlížeči.

JednotkySearchForm

V této třídě se řeší vyhledávání jednotek pomocí vyhledávacích formulářů navržených v programu `Qt Designer`. Vyhledávat lze podle čísla jednotky, domovního čísla, čísla parcely, ke které jednotka patří, listu vlastnictví nebo způsobu využití jednotky. Třída je potomkem `QWidget`.

LatexDocument

Třída `LatexDocument` obsahuje metody, pomocí nichž se vytváří části kódu, které jsou nakonec sestaveny jako *L^AT_EX* dokument. Takto sestavený dokument je poté vyexportován pomocí příslušné funkcionality modulu a je připravený pro export například od formátu *PDF* (`pdflatex`). Třída je potomkem `VfkDocument`.

MainApp

Třída `MainApp` je stěžejní třídou zásuvného modulu. Je potomkem tříd `QDockWidget`, `QMainWindow` a `Ui_MainApp`. Obsahuje metody pro načítání VFK souborů, nastavuje symbologii načtených vrstev, propojuje vyhledávání s mapovým oknem `QGIS`, stará

se o exporty do definovaných formátů. Ve třídě se také přidává funkcionality jednotlivým prvkům zásuvného modulu.

OpenThread

Tato třída se stará o načítání jednotlivých *VFK* souborů v separátním vlákne. Díky použití jiného vlákna pro načítání souborů, se práce se zásuvným modulem stává plynulejší. Třída dědí od třídy `QThread` dostupné v použité knihovně `PyQt4`. Obsahuje jedinou metodu (`run()`) pro spuštění separátního vlákna, jejíž předpis je dostupný v příloze č. B.3.

ParcelySearchForm

Třída `ParcelySearchForm` stejně jako předešlé třídy pro vyhledávání dědí od rodičovské třídy `QWidget`. Obsahuje metody, které pracují s navrženými grafickými prvky. Díky těmto metodám je možné vyhledávat parcely podle parcelního čísla, typu parcely, druhu pozemku nebo listu vlastnictví.

RichTextDocument

Třída se stará o tvorbu *HTML* dokumentu, který obsahuje výsledky vyhledávání a je zobrazen v prohlížeči dat. Třída je potomkem `VfkDocument`.

SearchFormController

Třída `SearchFormController` získává hodnoty ze vstupních formulářů, které jsou obsahem tříd pro vyhledávání (`BudovySearchForm`, `JednotkySearchForm`, `ParcelySearchForm`, `VlasniciSearchForm`). Z těchto vstupních hodnot poté sestavuje vyhledávací dotaz, který pomocí signálu `actionTriggered` emituje dále. Ukázka kódu pro sestavení vyhledávacího dotazu pro parcely je uvedena v příloze B.4.

VfkDocument

Jedná se o abstraktní třídu, ze které jsou odvozeny třídy pro sestavení dokumentů v různých formátech (`HtmlDocument`, `LatexDocument`, `RichTextDocument`). Jsou zde implementovány základní metody, které zajišťují správné sestavení dokumentu.

VfkPlugin

Tato třída implementuje zásuvný modul do programu QGIS. Jsou zde uvedeny metody, které vytvářejí instanci pluginu v programu QGIS po jeho instalaci, nebo metody, které tuto instanci ruší. Tato třída byla vygenerována automaticky při tvorbě základního schéma pluginu nástrojem *Plugin Builder* a později upravena dle konkrétních potřeb.

VfkTableModel

Tato třída je potomkem `QSqlQueryModel`, která poskytuje model pro čtení SQL dotazů. Třída komunikuje s připojenou databází, která je do třídy předávána pomocí argumentu `connectionName`. Ke komunikaci jsou používány standardní SQL dotazy, které jsou do databáze předávány metodou `self.setQuery()`.

VfkTextBrowser

`VfkTextBrowser` je potomkem třídy `QTextBrowser`. Třída byla stanovena jako „zástupná“ pro prvek `vfkBrowser` v okně zásuvného modulu. Tento prvek slouží jako prohlížeč dat.

VlastniciSearchForm

Jedná se o třídu, která slouží k vyhledávání vlastníků podle jména, typu (fyzická osoba, právnická osoba, společné jmění manželů), rodného čísla (případně IČO) nebo listu vlastnictví. Metody třídy získávají hodnoty vstupních polí / „checkboxů“.

Příloha B

Ukázky kódu

B.1 Konstruktor třídy BudovySearchForm

```
1 class BudovySearchForm(QWidget):
2
3     def __init__(self, parent=None):
4         super(BudovySearchForm, self).__init__(parent)
5
6         # Set up the user interface from Designer.
7         self.ui = Ui_BudovySearchForm()
8         self.ui.setupUi(self)
9
10        self._mZpusobVyuzitiModel = QAbstractItemModel
```

B.2 Metoda buildHtml

```
1 def buildHtml(self, document, taskMap):
2     """
3     :type document: VfkDocument
4     :type taskMap: dict
5     """
6     self._mCurrentPageParIds = []
7     self._mCurrentPageBudIds = []
8     self._mCurrentDefinitionPoint.first = ''
9     self._mCurrentDefinitionPoint.second = ''
10
11    self._mDocument = document
12    self._mDocument.header()
13
14    if taskMap["page"] == "help":
15        self.pageHelp()
16
17    if self._mHasConnection:
18        if taskMap["page"] == "tel":
19            self.pageTeleso(taskMap["id"])
20        elif taskMap["page"] == "par":
21            self.pageParcela(taskMap["id"])
22        elif taskMap["page"] == "bud":
23            self.pageBudova(taskMap["id"])
24        elif taskMap["page"] == "jed":
25            self.pageJednotka(taskMap["id"])
26        elif taskMap["page"] == "opsub":
```



```
27     self.pageOpravnenySubjekt(taskMap["id"])
28     elif taskMap["page"] == "seznam":
29         if taskMap["type"] == "id":
30             if "parcely" in taskMap:
31                 self.pageSeznamParcel(taskMap["parcely"]
32                                     .split(","))
33             if "budovy" in taskMap:
34                 self.pageSeznamBudov(taskMap["budovy"]
35                                     .split(","))
36         elif taskMap["type"] == "string":
37             if "opsub" in taskMap:
38                 self.pageSeznamOsob(taskMap['opsub']
39                                     .split(","))
40     elif taskMap["page"] == "search":
41         if taskMap["type"] == "vlastnici":
42             self.pageSearchVlastnici(
43                 taskMap["jmeno"],
44                 taskMap["rcIco"],
45                 taskMap["sjm"],
46                 taskMap["opo"],
47                 taskMap["ofo"],
48                 taskMap["lv"])
49         elif taskMap["type"] == "parcely":
50             self.pageSearchParcely(
51                 taskMap["parcelniCislo"],
52                 taskMap["typ"],
53                 taskMap["druh"],
54                 taskMap["lv"])
55         elif taskMap["type"] == "budovy":
56             self.pageSearchBudovy(
57                 taskMap["domovniCislo"],
58                 taskMap["naParcele"],
59                 taskMap["zpusobVyuziti"],
60                 taskMap["lv"])
61         elif taskMap["type"] == "jednotky":
62             self.pageSearchJednotky(
63                 taskMap["cisloJednotky"],
64                 taskMap["domovniCislo"],
65                 taskMap["naParcele"],
66                 taskMap["zpusobVyuziti"],
67                 taskMap["lv"])
68     self.__mDocument.footer()
69     return
```

B.3 Načtení souborů v separátním vlákně

```
1     def run(self):
2         # load all VFK files
3         for i, vfkFile in enumerate(self.vfk_files):
4             self.working.emit(vfkFile)
5             self.nextLayer = True
6
7         while self.nextLayer:
8             self.sleep(1)
```

B.4 Vyhledání parcel – ukázka tvorby URL

```
1     def __searchParcely(self):
2         """
3         Method creates and emits query for searching of plots.
4         """
```



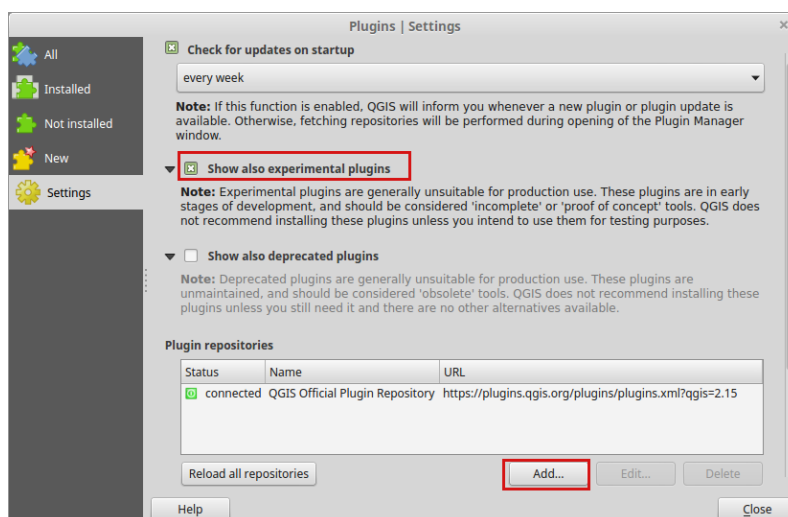
```
5 parcelniCislo = self.__forms.parcely.parcelniCislo()
6 typ = int(self.__forms.parcely.typParcely())
7 druh = self.__forms.parcely.druhPozemkuKod()
8 lv = self.__forms.parcely.lv()
9
10 # build url
11 url = QUrl(u"showText?page=search&type=parcely&parcelniCislo={}&typ={}&druh={}&lv={}"
12         .format(parcelniCislo, typ, druh, lv))
13 self.actionTriggered.emit(url)
```

Příloha C

Uživatelská příručka

C.1 Instalace a spuštění zásuvného modulu

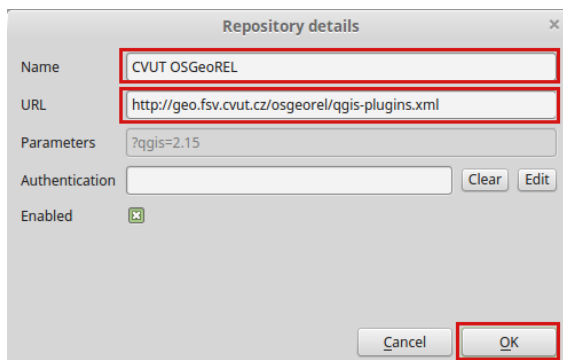
Plugin v současné době není dostupný v oficiálním repozitáři QGIS. Pro instalaci pluginu je potřeba do QGIS přidat nový repozitář spravovaný organizací OSGeoREL¹, pod kterou je tento plugin vyvíjen. Repozitář je dostupný na adrese <http://geo.fsv.cvut.cz/osgeorel/qgis-plugins.xml>, viz obr. C.2. Modul je šířený jako experimentální, proto musí být tato volba zohledněna při jeho instalaci, viz obr. C.1. Okno pro zadání repozitáře vyvoláme v záložce *Plugins* → *Manage and install plugins*.



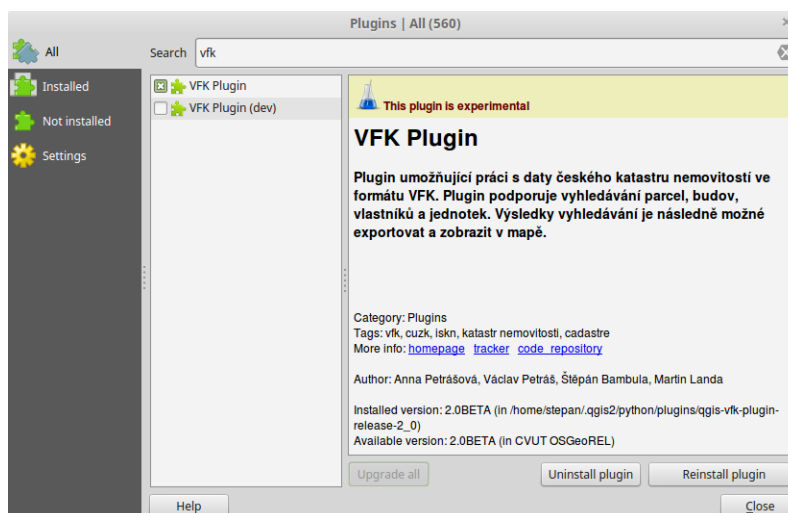
Obrázek C.1: Přidání repozitáře QGIS

Po přidání repozitáře a jeho aktualizaci je zásuvný modul dohledatelný standardním způsobem pod názvem **VFK Plugin**, viz obr. C.3.

¹<http://geomatics.fsv.cvut.cz/research/osgeorel>



Obrázek C.2: Přidání repositáře QGIS – detail



Obrázek C.3: Vyhledání zásuvného modulu

Po správné instalaci zásuvného modulu se do nástrojové lišty přidá jeho ikonka (viz obr. C.4). Plugin je poté dostupný po kliknutí na tuto ikonku nebo z menu *Plugins* → *VFK* → *Otevřít prohlížeč VFK*.

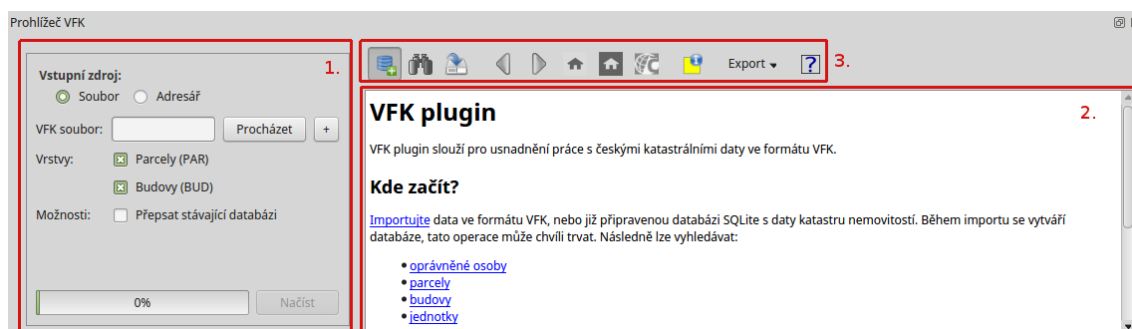


Obrázek C.4: VFK Plugin – ikona

Uživatelská přívětivost používání zásuvného modulu je zvýšena „dokovatelností“ samotného okna pluginu. Modul je primárně ukotven k horní liště okna QGIS, ale je možné ho ukotvit i k ostatním okrajům. Okno modulu lze samozřejmě používat i samostatně (neukotvené).

C.2 Rozložení zásuvného modulu

Okno zásuvného modulu se skládá ze tří hlavních částí. Vlevo se nachází hlavní panel nástrojů (obr C.5 prvek č. 1) pro přepínání oken pro import souborů ve formátu *.vfk*, *.db* a pro vyhledávání. Naprostou většinu pravé části tvoří prohlížeč dat (obr C.5 prvek č. 2), ve kterém se zobrazují výsledky vyhledávání a nápověda. Nad ním je umístěna nástrojová lišta (obr C.5 prvek č. 3), která obsahuje nástroje pro práci se zásuvným modulem.



Obrázek C.5: VFK Plugin – rozložení prvků

C.3 Načítání dat

Pomocí zásuvného modulu je možné načíst soubor VFK nebo soubor databáze SQLite, ve které jsou VFK data uložena.

Funkcionalita pro načítání více souborů (jednotlivě nebo z adresáře) a načítání dat z SQLite databáze je do zásuvného modulu přidána pouze v případě, kdy je na počítači uživatele dostupná knihovna GDAL ve verzi 2.2.0 a vyšší.

C.3.1 Načítání ze souborů

Zásuvný modul podporuje načítání souborů ve formátu VFK nebo databáze SQLite. Cesta ke vstupním souborům je zadána pomocí tlačítka „Procházet“ v nabídce *Import VFK*.

Je podporováno načítání více souborů do jedné databáze. Formulář pro načtení dalšího VFK souboru je vygenerován po stisku tlačítka „+“. Cesta k novému souboru se zadává opět po stisku tlačítka „Procházet“.

C.3.2 Načítání z adresáře

Při čtení z adresáře uživatel musí zadat cestu k požadovanému adresáři obdobně, jako je tomu při čtení ze souborů. Při této volbě jsou načteny veškeré soubory VFK, které obsahuje zadaný adresář (včetně jeho podadresářů).

C.3.3 Možnosti načítání

V okně pro načítání dat je možné zvolit, které vrstvy se budou importovat (PAR, BUD). V případě, kdy není zatrženo ani jedno pole, tak se načtou pouze popisná data. Bude tedy dostupné pouze vyhledávání bez interakce s mapovým oknem.

Jestliže je na vstupu zadán soubor ve formátu VFK, tak je na pozadí vytvořena databáze SQLite, do které jsou data z VFK souboru načteny. Tato databáze je pojmenována `jmeno_prvniho_souboru_stav.db` pro případ, kdy načtený soubor obsahuje stavová data, nebo `jmeno_prvniho_souboru_zmeny.db`, když jsou v prvním souboru uložena změnová data. Databáze je uložena o adresář výše, než první soubor VFK.

V témže okně je možné zvolit možnost „Přepsat stávající databázi“, která umožní znovunačtení dat VFK do SQLite databáze. Při této možnosti je původní databáze smazána a je vytvořena znovu dle nově načteného souboru VFK.

Upozornění: Není umožněna kombinace více typů datových zdrojů!

C.4 Vyhledávání

Po úspěšném načtení dat je zpřístupněno vyhledávání. Výsledky vyhledávání se zobrazují ve vedlejším okně, prohlížeči dat. Tato data lze interaktivně procházet, podobně jako je tomu ve webovém prohlížeči, jelikož jsou ukládána ve formě HTML stránek. Pro navigaci mezi jednotlivými prvky jsou používány hypertextové odkazy. Obsah okna prohlížeče je ukládán do historie, což usnadňuje uživateli vyhledávání stále se opakujících informací (není potřeba opakovaně provádět stejný databázový dotaz). Historii lze procházet pomocí tlačítek „Vpřed“ a „Zpět“.

Pro zjištění aktuálního stavu o dané nemovitosti je zde implementováno propojení s aplikací **Nahlížení do katastru nemovitostí**².

Zásuvný modul a mapové okno QGIS jsou navzájem propojené. Je tedy možné v mapovém okně zobrazit polohu vyhledaných prvků a opačně vyhledat informace o prvcích, které byly vybrány pomocí nástroje výběru QGIS právě v mapovém okně (při výběru prvků z mapového okna je potřeba aktivovat tlačítko „Aktivovat / deaktivovat zobrazení informací o vybraných nemovitostech“).

²<http://nahlizeniidokn.cuzk.cz/>

Vyhledávat lze podle následujících prvků:

- vlastníků,
- parcel,
- budov,
- jednotek.

(a) Vyhledávání vlastníků

(b) Vyhledávání parcel

Obrázek C.6: Vyhledávací formuláře 1/2

(a) Vyhledávání budov

(b) Vyhledávání jednotek

Obrázek C.7: Vyhledávací formuláře 2/2

C.5 Export vyhledaných informací

Veškeré vyhledané informace, které jsou zobrazeny v prohlížeči dat, je možné uložit. Export může být proveden do souboru v následujících formátech:

- **HTML:** Umožňuje následné zobrazení ve webovém prohlížeči, případně import do textového procesoru
- **Zdrojový kód \LaTeX :** Umožňuje vytvoření PDF nebo PS.

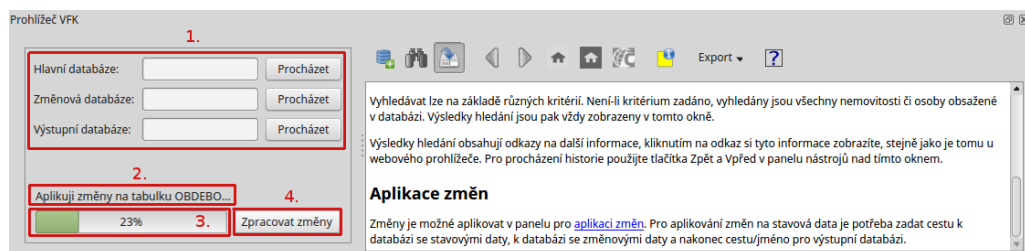
C.6 Zpracování změn

Funkcionalita pro zpracování změn je dostupná pod příslušnou ikonou (obr C.8). Po stisku této ikony je vygenerováno nové okno s následujícími prvky:



Obrázek C.8: VFK Plugin – ikona pro zpracování změn

1. **Formuláře:** K dispozici jsou tři formuláře pro zadání cesty ke vstupní databázi se stavovými daty, k databázi se změnovými daty a databázi výstupní, do které budou uložena stavová data s aplikovanými změnami.
2. **Popisek průběhu zpracování:** V tomto popisku jsou zobrazovány informace o průběhu zpracování změn.
3. **Grafický indikátor průběhu:** Zde je průběh zpracování změn vyjádřen procentuálně a graficky.
4. **Tlačítko pro zpracování:** Po stisku tohoto tlačítka jsou změny aplikovány do databáze se stavovými daty.



Obrázek C.9: VFK Plugin – okno pro zpracování změn

Postup zpracování změn je následující. Nejprve jsou načteny pomocí tlačítek „Procházet“ databáze se stavovými daty a databáze se změnami. Dále je potřeba zadat cestu k nově vytvořené databázi, do které budou změny aplikovány. Po zadání těchto informací je zpřístupněno tlačítko „Zpracovat změny“. Změny jsou na stavová data aplikovány po jeho stisknutí.

Funkcionalita pro aplikaci změn obsažených v databázi SQLite na data obsažená v databázi se stavovými daty je do zásuvného modulu přidána pouze v případě, kdy je na počítači uživatele dostupná knihovna GDAL ve verzi 2.2.0 a vyšší. V opačném případě je ikonka pro zpracování změn neaktivní.

Příloha D

Obsah přiloženého CD

Přiložené CD obsahuje zdrojové kódy zásuvného modulu, text diplomové práce a „patch“ knihovny GDAL vytvořený oproti verzi 2.2.0. Struktura obsahu CD je uvedena níže.

```
/
├── src/
│   ├── data/
│   ├── applyChanges.py
│   ├── mainApp.py
│   ├── vfkPlugin.py
│   ├── resources_rc.py
│   └── :
├── dp/
│   ├── text/
│   │   ├── images/
│   │   ├── citace.bib
│   │   ├── Makefile
│   │   ├── stepan-bambula-dp-2016.tex
│   │   ├── stepan-bambula-dp-2016.pdf
│   │   └── :
│   ├── zadani/
│   │   ├── zadanidp.doc
│   │   └── zadanidp.pdf
│   ├── .gitignore
│   └── README.md
└── gdal/
    └── gdal-patch-vfk.diff
```