

**ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE
FAKULTA STAVEBNÍ**

Katedra mapování a kartografie



**Webový systém pro správu mapových
aplikací digitálních modelů terénu**

BAKALÁŘSKÁ PRÁCE

Autor práce:

ADAM POLÍVKA

Vedoucí práce:

ING. JIŘÍ CAJTHAML, PH. D.

Praha, 2010

PROHLÁŠENÍ

Prohlašuji, že jsem svou bakalářskou práci na téma „*Webový systém pro správu mapových aplikací digitálních modelů terénu*“ vypracoval samostatně a použil jsem pouze literaturu (podklady, projekty, SW atd.) uvedené v příloženém seznamu.

Nemám závažný důvod proti užití tohoto školního díla ve smyslu § 60 zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a změně některých zákonů (autorský zákon).

V Praze dne _____

vlastnoruční podpis autora

PODĚKOVÁNÍ

Tímto bych chtěl poděkovat vedoucímu bakalářské práce Ing. Jiřímu Cajthamlovi, Ph. D. za cenné rady, jeho důvěru a za to, že mně umožnil pracovat na v praxi využitelném a pro mě zajímavém projektu. Mé poděkování patří také mé rodině za podporu při studiu.

WEBOVÝ SYSTÉM PRO SPRÁVU MAPOVÝCH APLIKACÍ DIGITÁLNÍCH MODELŮ TERÉNU

SOUHRN

Předmětem této bakalářské práce je popis vytvořeného webového systému na správu mapových projektů. Aplikace je navržena na síťové architektuře klient-server. Tématem první části je popis použitých technologií na straně serveru – programovací jazyk Python, databázový systém PostgreSQL, geografický informační systém GRASS, mapový server pro publikování geografických dat MapServer a aplikace pro generování dlaždic TileCache. Druhá kapitola je zaměřena na použité technologie na straně webového klienta – javascriptový framework Ext JS a javascriptové knihovny Mapfish a OpenLayers. Třetí část se orientuje na popis ovládání vlastní aplikace. V závěru se snažím zhodnotit vytvořenou aplikaci.

KLÍČOVÁ SLOVA

- AJAX
- Analýza rastrových geoprostorových dat
- Digitální model terénu
- Ext JS
- GRASS GIS
- Mapfish
- Mapserver
- OpenLayers
- PostgreSQL
- Python
- RIA
- Tilecache
- WMS

WEB SYSTEM FOR DIGITAL TERRAIN MODELS MAP APPLICATIONS MANAGEMENT

SUMMARY

The subject of this bachelor's thesis is a description of web-based application for management of digital model terrain models map projects. The application is designed in the client-server model. The first chapter is focused on the used server-side technologies – such as programming language Python, database management system PostgreSQL, geographic information system GRASS, application for publishing geospatial data Mapserver and web map tile catching mechanism TileCache. The second part deals with the used client-side technologies – mainly framework for creating rich internet applications Ext and javascript libraries for maps publishing OpenLayers and MapFish. The third section describes the application and its control. At the end, I try to evaluate practical usability of this web application.

KEY WORDS

- AJAX
- Analysis of raster geospatial data
- Digital model terrain
- Ext JS
- GRASS GIS
- MapFish
- Mapserver
- OpenLayers
- PostgreSQL
- Python
- RIA
- TileCache
- WMS

OBSAH

Úvod.....	1
1. POUŽITÉ TECHNOLOGIE A SOFTWARE NA STRANĚ SERVERU.....	2
1.1 Úvod.....	2
1.2 Webový server Apache.....	2
1.3 Databázový systém PostgreSQL.....	3
1.4 Python.....	4
1.4.1 Práce se soubory	5
1.4.2 Modul pycogp2	6
1.4.3 Python a cgi.....	7
1.5 GRASS	8
1.5.1 Práce s rastrovými daty.....	9
1.5.2 Práce s vektorovými daty	10
1.6 MapServer.....	12
1.7 TileCache	15
1.7.1 Konfigurace TileCache	16
1.7.2 Seeding	17
2. POUŽITÉ TECHNOLOGIE NA STRANĚ KLIENTA	18
2.1 Úvod.....	18
2.2 JavaScript.....	18
2.3 Ext JS.....	19
2.3.1 Práce s AJAX.....	20
2.3.2 Práce s DOM	20
2.3.3 Užitečné funkce	21
2.3.4 GUI objekty.....	22
2.4 OpenLayers.....	24
2.5 MapFish Client.....	25

3. POPIS VLASTNÍ APLIKACE	26
3.1 Úvod.....	26
3.2 Ovládání.....	26
3.2.1 Přihlášení.....	26
3.2.2 Po přihlášení	26
3.2.3 Editace	27
3.2.4 Vizualizace	29
ZÁVĚR	32
POUŽITÉ ZDROJE INFORMACÍ	33
POUŽITÉ ZKRATKY A TERMÍNY	35
SEZNAM OBRÁZKŮ	36

Úvod

Geografický informační systém (GIS) je pro získávání, ukládání, správu a vizualizaci geografických dat nepostradatelná aplikace. Tradiční desktopový GIS je mocný nástroj schopný provádět složité analýzy prostorových dat a umožňující nám odpovídat na nejzákladnější otázky týkající se objektů v prostoru. Velmi komplexní otázky si však většinou pokládají pouze odborníci, kteří mají s užíváním těchto systémů značné zkušenosti. Běžný uživatel GIS si častěji klade mnohem jednodušší otázky či otázky úzce zaměřené na svůj obor, při jejichž řešení využije pouze malé množství funkcí těchto systémů. Velké množství ovládacích prvků k funkcím, které uživatel nevyužívá, mu pak pouze znepríjemňují práci se systémem. Systém také využívá více systémových prostředků na počítači uživatele než je nezbytné.

V dnešní době díky rychlému rozvoji internetu a WWW vzniká stále více aplikací založených na síťové architektuře klient-server. Tento trend se samozřejmě projevil i ve vývoji GIS. Výhody webového GIS spočívají v tom, že data jsou snadno přístupná více klientům z jednoho místa a klientské aplikace je jednodušší přizpůsobit potřebám uživatele. Webové aplikace nejsou závislé na operačním systému.

Moje práce si klade za cíl navrhnout a vytvořit interaktivní mapovou webovou aplikaci navrženou pro konkrétní využití – vizualizaci změn digitálního modelu terénu v různých časových etapách. Aplikace má být navržena pro firmu zabývající se laserovým skenováním a má sloužit jejím zákazníkům z oblasti těžby surovin k vyhodnocení jednotlivých etap těžby.

Aplikace by měla uživatelům umožnit jednoduše a rychle nastavit vlastnosti projektu. To znamená přidávat nové vrstvy digitálních modelů terénu, připojovat WMS vrstvy (například katastrální mapu, ortofotomapu) a přidávat dokumenty týkající se projektu. Systém by měl ukládat informace o projektech do databáze.

Primárním cílem aplikace je v rámci jednoho projektu vizualizace výškových modelů v jednotlivých etapách i jejich rozdílů, aby bylo možné sledovat změny. Etapy by měly být snadno přepínatelné. Aplikaci by měla povolit měnit u každé vrstvy volbu, jestli se má vrstva zobrazovat, a umožnit měnit pořadí zobrazovaných vrstev.

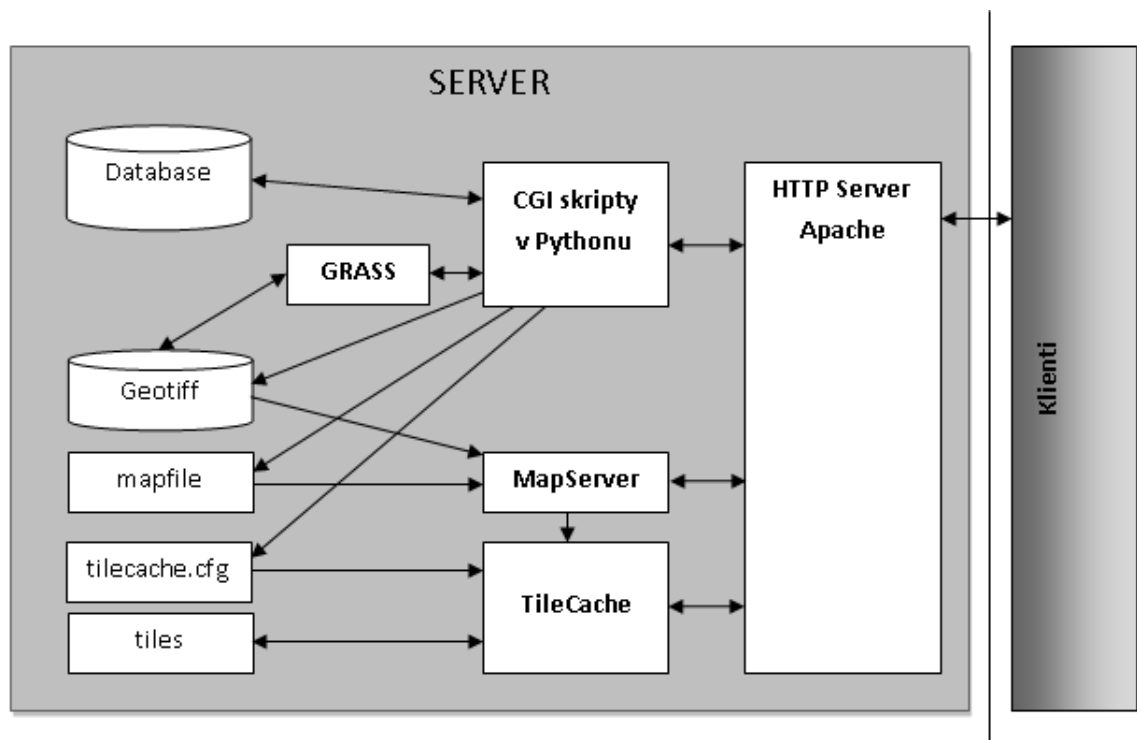
1. POUŽITÉ TECHNOLOGIE A SOFTWARE NA STRANĚ SERVERU

1.1 Úvod

Aplikace běží na fakultním serveru katedry mapování a kartografie, kde je nainstalována distribuce Ubuntu operačního systému GNU/Linux. Na serveru byl využit tento software:

- Apache/2.2.8 (Ubuntu)
- PostgreSQL 8.1.10
- Python 2.5.2
- GRASS 6.4.0RC5
- Mapserver 5.4.2-1
- Tilecache 2.10

Obrázek 1: Architektura serveru



1.2 Webový server Apache

Apache HTTP Server (zkráceně Apache) je open-source webový server vydávaný pod licencí Apache Licence¹. Apache je možné nainstalovat na většinu používaných operačních

¹ Apache Licence vytvořena Apache Software Foundation, <http://www.apache.org/>

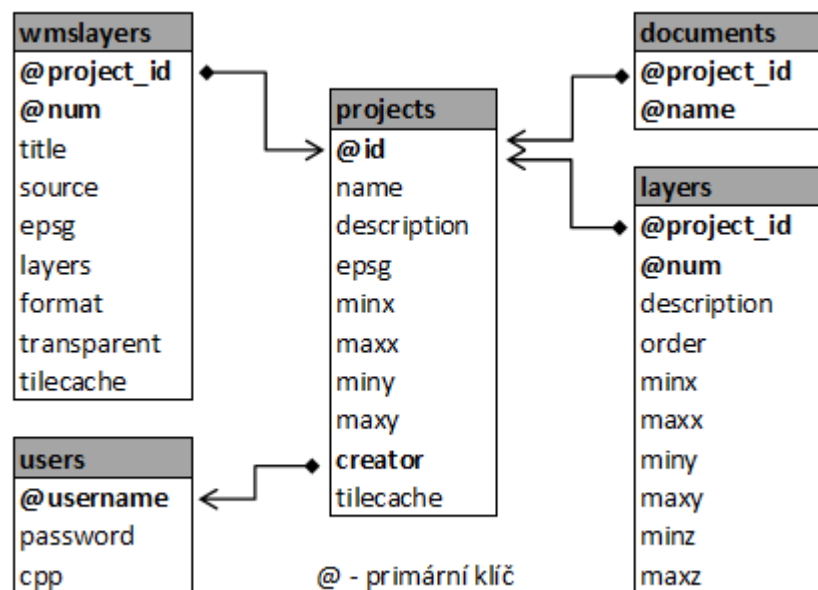
systémů včetně GNU/Linuxu, Unixu a od něj odvozených operačních systémů a MS Windows. Pro své dobré vlastnosti – spolehlivost, stabilitu, bezpečnost - se stal velmi populární a dnes se jedná o nejpoužívanější webový server na Internetu.² Více informací na oficiálních stránkách projektu <http://httpd.apache.org/>.

1.3 Databázový systém PostgreSQL

PostgreSQL je objektově-relační databázový systém s otevřeným zdrojovým kódem, který je šířen pod liberální licenci typu MIT.³ Nativně běží na všech rozšířených operačních systémech jako například GNU/Linux, Unix, BSD, Solaris, Mac OS X či Windows.

Výkonnostně a v mnoha jiných ohledech se PostgreSQL často vyrovná komerčním databázovým systémům. Při vývoji systému je snaha o respektování a implementaci ANSI SQL⁴. Na stránkách projektu je dostupná kvalitně zpracovaná dokumentace.

Obrázek 2: Základní schéma databáze



² V únoru 2010 činil podíl na trhu podle výzkumu společnosti Netcraft 54%, aktivních stránek 52%, zdroj: http://news.netcraft.com/archives/2010/02/22/february_2010_web_server_survey.html

³ Text PostgreSQL License dostupný z WWW: <http://www.postgresql.org/about/licence>

⁴ Zdroj: <http://www.postgresql.org/docs/8.1/static/features.html>

Obrázek 3: Logo PostgreSQL

Zdroj: <http://www.postgres.cz/images/8/83/Logo180.jpg>

1.4 Python

Python je vysokoúrovňový programovací jazyk navržený s důrazem na vysokou produktivitu programátora a čitelnost kódu, které je dosaženo striktní syntaxí – především odsazováním. Python se řadí mezi objektově orientované jazyky. Současně je však navržen jako paradigmální jazyk, umožňující používat různé styly programování – objektově-orientovaný, procedurální či funkcionální. Právě díky možnosti používání různých stylů programování je Python použitelný jak na psaní malých skriptů o několika řádcích, tak pro psaní velkých projektů. Python je interpretovaný jazyk, což znamená, že zdrojové kódy se překládají do strojového kódu až při jejich spuštění. Interpret jazyka je multiplatformní, což umožňuje stejný skript spustit na libovolném podporovaném operačním systému. Pro své kladné vlastnosti se Python stal mezi vývojáři velmi populární a v poslední době se řadí mezi nepoužívanější programovací jazyky.⁵

Obrázek 4: Logo projektu Python

Zdroj: <http://python.org/>

Většina kódů je vydána pod Python licenci⁶, která je plně kompatibilní s GPL. Tuto licenci vytvořila nezisková organizace PSF⁷. Z licence plyne, že Python lze svobodně využít,

⁵ Podle indexu společnosti Tiobe je Python 7. nepoužívanější programovací jazyk, duben 2010, zdroj: <http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html>

⁶ Plné znění Python Licence v ang. jaz.: <http://www.python.org/download/releases/2.5.2/license/>

⁷ Python Software Foundation, stránky nadace: <http://www.python.org/psf/>

dokonce i pro komerční užití, bez nutnosti placení poplatků PFS. Tato licence je zařazena na seznamu open-source licencí⁸ uznané Open Source Initiative⁹.

Přestože Python obsahuje rozsáhlou standardní knihovnu, je navržen tak, aby mohl být snadno rozšiřován pomocí modulů napsaných v jiných jazycích (např. C, C++, Java). Existuje několik implementací Pythonu psaných v různých jazycích – CPython (jazyk C), Jython (Java), IronPython (C#), PyPy(Python). Nejrozšířenější implementací jazyka Python je CPython, který je napsán v jazyku C.

Python používá dynamickou typovou kontrolu, z čehož plyne, že proměnné mohou během vykonávání programu nabývat hodnot o různých datových typech. Při operacích je však uplatňována silná typová kontrola.

Ukázka kódu dokazující, že Python má silnou typovou kontrolu

```
>>> 1+"1"
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
TypeError: unsupported operand type(s) for +: 'int' and 'str'
```

Zajímavou vlastností jazyka je tzv. „duck typing“ Jedná se o způsob zápisu kódu, který vychází z dynamické kontroly a umožňuje například psát funkce bez znalosti typů vstupních proměnných.

Jednoduchá ukázka duck typing

```
>>> def secti(a,b): return a+b
>>> secti(1,2)
3
>>> secti('Pyth', 'on')
'Python'
```

Jako vstupní parametry funkce pak můžeme zadat proměnné jakéhokoli typu, dokud podporují vnitřní metody. Z toho vyplývá další vlastnost jazyka – polymorfismus bez dědičnosti.

1.4.1 Práce se soubory

Objekty pracující se soubory využívají implementaci knihovny stdio jazyka C. Základní funkce k vytvoření objektu `file` je metoda `open()`:

⁸ Seznam open-source licencí: <http://www.opensource.org/licenses/>

⁹ Open Source Initiative(OSI) je organizace věnující se open-source softwaru. Oficiální stránky: <http://www.opensource.org/>

Vytvoření objektu typu file

```
soubor = open('soubor.txt', 'r')
```

První parametr funkce značí název souboru a druhý mode: 'r' pro čtení, 'w' pro přepsání, 'a' pro připsání na konec souboru. Objekt soubor pak má metody pro čtení: `read()`, `readline()` nebo `readlines()`. Pokud je soubor otevřen pro zápis je možné použít metodu `write()`. Po ukončení práce s objektem se soubor uzavírá metodou `close()`.

Metody objektu typu file ke čtení

```
soubor.read()      # vrátí celý obsah souboru
soubor.readline()  # vrátí první nepřečtenou řádku
soubor.readlines() # vrátí list s řádky
```

Zápis do souboru

```
soubor.write('Ahoj světe')
```

Ukončení práce se souborem

```
soubor.close()
```

1.4.2 Modul psycopg2

Pro práci s datlovým systémem PostgreSQL, který byl využit pro ukládání informací o projektech, slouží modul `psycopg2`. Nejprve je třeba modul načíst a pro spojení s databází vytvořit objekty typu `connection` a `cursor`:

Import modulu

```
import psycopg2
```

Spojení s databází - vytvoření objektu typu connection a cursor

```
conn = psycopg2.connect("""user='uzivatel' host='localhost'
                           dbname='dbname' password='heslo' """)
cur = conn.cursor()
```

Příkazy se provádí voláním metody `execute()` objektu typu `cursor`. K výsledku se pak dostaneme například zavoláním metody `fetchall()`, která vrátí výsledek jako seznam záznamů.

Ukázka jednoduchého SELECT dotazu do databáze a výpis

```
cur.execute("SELECT id,name FROM controlsystem.projects")
records = cur.fetchall()
for record in records:
    print "id: %d,name: %s"%(record[0],record[1])
```

Pokud v databázi chceme provést změnu (tj. vykonat SQL příkazy jako INSERT, DELETE, UPDATE), příkaz `execute()` se neprovede, dokud zápis neschválíme (např. metodou `commit()`) nebo databázi neuzavřeme (metoda `close()`).

Zápis do databáze se provede až po zavolání metody commit

```
cur.execute("INSERT INTO potraviny(položka) VALUES('sýr')")
conn.commit()
```

1.4.3 Python a cgi

Python obsahuje modul `cgi`, který poskytuje jednoduché rozhraní pro skripty v Pythonu a obsahuje řadu užitečných nástrojů pro ladění CGI skriptů.

CGI skript je spuštěn přes HTTP server, obvykle ke zpracování uživatelského vstupu zadaného přes HTML element `<form>`. Obvykle jsou CGI skripty umístěny na serveru ve speciální složce `cgi-bin`. HTTP server umístí všechny druhy informací o požadavku (požadované url, query string, hostname klienta a další informace) do prostředí shellu, spustí skript a vyšle výstup skriptu zpět klientovi.

První výstup, který musí cgi skript vypsát je informace o Content-type

```
print "Content-Type: text/html\n"
```

ukázka zpracování přijatých parametrů

```
import cgi
form = cgi.FieldStorage()
if "jmeno" in form and form.has_value("prijmeni"):
    jmeno = form["jmeno"].value
    prijmeni = form["prijmeni"].value
else:
    print "formular neni vyplnen"
    return
print "vase cele jmeno je",jmeno,prijmeni
```

Ukázka uploadu souboru na server

```
fileitem = form["soubor"]
if fileitem.file:
    fout = file(os.path.join('/tmp/', fileitem.name), 'wb');
    while 1:
        chunk = fileitem.file.read(100000);
        if not chunk:
            break
        fout.write(chunk)
    fout.close()
```

1.5 GRASS

Geografický informační systém GRASS (akronym anglického názvu Geographic Resources Analysis Support System) je svobodný software s otevřeným kódem používaný pro správu a analýzu geoprostorových dat, tvorbu a úpravu rastrových obrázků a map, vizualizaci a prostorové modelování. Na vývoji GRASSu se od roku 1982 podílela řada vládních agentur Spojených států amerických, světových univerzit a soukromých společností. V dnešní době je oficiálním projektem Open Source Geospatial Foundation vydávaným pod licencí GNU GPL a může být nainstalován na mnoho operačních systémů včetně těch nejrozšířenějších: MS Windows, Mac OS X a GNU/Linux.

Obrázek 5: Logo GRASS GIS

Zdroj: http://grass.osgeo.org/images/grasslogo_vector_small.png

Systém je ovládán voláním specifických příkazů, označovaných jako moduly, v upraveném prostředí UNIX shellu. V prostředí jsou zaznamenány stavové parametry (například o geografickém regionu, mapové projekci, či masce). Všechny moduly pak před provedením načítají stavové parametry doplněné o další parametry zadané při volání (například vstupní a výstupní data, hodnoty užívané k výpočtům). Právě pro dobrou podporu skriptování byl GRASS zvolen jako nástroj pro analýzu a zpracování geografických dat v této aplikaci.

K ručnímu zadávání příkazů v shellu existuje pro uživatele alternativa v podobě ovládání přes grafické uživatelské prostředí. Od verze 6.4 GRASS používá wxGUI, napsané v programovacím jazyce Python a grafické knihovně wxPython.

Vzhledem k tomu, že aplikace využívá GRASS na serveru, budeme se zde věnovat více popisu skriptování přes shell a modulům, kterých bylo v aplikaci použito. GRASS 6.4 totiž obsahuje celkem více než 300 modulů.

1.5.1 Práce s rastrovými daty

Rastrové mapy jsou datové vrstvy složené z buněk (cell) uspořádaných ve čtvercové síti. Mají určitý počet řad a sloupců. Každá buňka obsahuje hodnotu nebo hodnotu null. Před zpracováváním je třeba importovat data do nativního formátu GRASSu.

1.5.1.1 Import a export rastrových dat

K importu rastrových dat slouží různé moduly pracující s odlišnými vstupními daty (r.in.arc, r.in.ascii, r.in.aster, r.in.bin, r.in.gdal, r.in.gridatb, r.in.poly, r.in.srtm, r.in.wms, r.in.xyz). K importu dat ve formátu GeoTiff, se kterým vytvořená aplikace pracuje, se používá modul r.in.gdal, který využívá knihovnu GDAL podporující mnoho rastrových formátů¹⁰, včetně formátu GeoTiff.

Import rastrových dat do GRASSu

```
r.in.gdal input=<vstupni.tif> output=<importovana_vrstva>
```

Pro export existuje stejně jako pro import více modulů, které jsou tentokrát označeny počátečními znaky „r.out“. K exportu do formátu GeoTiff můžeme opět využít modul pracující s knihovnou GDAL.

Export dat (Modul r.out.gdal defaultně exportuje do formátu GeoTiff)

```
r.out.gdal input=<vstupni_vrstva> output=<vystupni.tif> type=Int16
```

1.5.1.2 Informace o vrstvě

K dispozici je celá řada příkazů sloužících k získání informací o vrstvě a jejích statistikách:

- `r.info` – zobrazí obecné informace o vrstvě
- `r.univar` – vypočte některé statistiky vrstvy
- `r.stats` – vypočte plošné statistiky

¹⁰ Seznam podporovaných formátů knihovny GDAL: http://www.gdal.org/formats_list.html

- `r.report` – vypíše report se statistikami vrstvy
- `r.sum` – sečte hodnoty buněk
- `r.statistics` – vypočte statistiky podle kategorií

1.5.1.3 Region a maska

Při práci s rastry je třeba si dát pozor na to, že GRASS na vstupu zpracovává pouze data, která se nacházejí v aktuálním regionu a která odpovídají masce. Region se nastavuje příkazem `g.region`.

Nastavení regionu podle rastrové vrstvy

```
g.region rast=<vrstva>
```

1.5.1.4 Rastrová aritmetika

V aplikaci bylo třeba vytvořit skript, který vytvoří rozdílovou vrstvu mezi dvěma vrstvami. K aritmetickým operacím nad rastrovými vrstvami slouží modul `r.mapcalc`. Jak již bylo napsáno, před samostatným výpočtem je třeba mít správně nastaven region a odstraněnou masku. U výškových dat se navíc často stává, že hodnota 0 u vstupních vrstev odpovídá hodnotě null. Při jejich rozdílu jsou pak v buňkách, kde jedna z vrstev nebyla definována, vypočtena nesmyslná čísla. Proto je před výpočtem nezbytné buňky s hodnotou 0 převést na hodnoty null.

Změna hodnoty na null

```
r.null map=vrstva1 setnull=0
```

Rozdíl dvou vrstev

```
r.mapcalc "rozdil=vrstva1-vrstva2"
```

1.5.2 Práce s vektorovými daty

GRASS podporuje vektorová data dvojrozměrná a trojrozměrná. Avšak trojrozměrný model a funkce ještě nejsou plně implementovány. Model vektorových dat je topologický. To znamená, že sousedící objekty v jedné vektorové mapě jsou propojeny svou hranicí, která je ve vrstvě zaznamenána pouze jednou. Naopak v netopologickém GISu jestliže dva plošné objekty sdílejí jednu hranici, tato hranice by byla v modelu uložena dvakrát. V jedné vrstvě mohou být objekty různého typu.

GRASS definuje následující vektorové objekty:

- P - Point – bod
- L - Line – orientovaná řada propojených vrcholů
- B - Boundary – hraniční linie plochy
- C – Centroid – bod uvnitř plochy
- F – Face – 3D plocha
- K – Kernel – 3D centroid
- A – Area – objekt definovaný hranicí a centroidem

1.5.2.1 Import vektorových dat

Pro import a export vektorových dat slouží stejně jako u rastrových dat více modulů rozdělených podle vstupního formátu. Moduly `r.in.ogr/r.out.ogr` využívající drivery knihovny OGR nabízejí rozhraní pro mnoho rozdílných vektorových formátů¹¹. Pro načítání dat z databáze slouží modul `v.in.db`, který dokáže pracovat například s PostgreSQL-PostGIS¹² či SQLite-Spatialite¹³. Pro speciální případy lze využít moduly `v.in.ascii` (resp. `v.out.ascii`) pracující s obyčejným textovým souborem obsahujícím geometrii a atributy. Tohoto formátu bylo využito pro import hranice označeného území, ve kterém uživatel chce spočítat kubaturu.

Ukázka dat ve formátu ASCII (Polygon o 4 bodech s centroidem)

```
B 4
-513171.9375 -1144867.75
-513178.8125 -1144882.125
-513163.8125 -1144895.25
-513171.9375 -1144867.75
C 1 1
-513171.9375 -1144867.75
```

Protože GRASS využívá topologický model vektorových vrstev, musí mít plošné objekty definovanou nejen svou hranici, ale i vnitřní bod - centroid. Pokud by plošný objekt neobsahoval centroid, systém by s ním nedokázal správně pracovat – tj. například vypočítat plochu objektu.

¹¹ Seznam podporovaných formátů knihovny OGR: http://www.gdal.org/ogr/ogr_formats.html

¹² Rozšíření ORDBS PostgreSQL o geografické objekty a funkce
stránky projektu: <http://postgis.refrations.net/>

¹³ Rozšíření databázového systému SQLite o geografickou funkcionalitu
stránky projektu: <http://www.gaia-gis.it/spatialite/>

Import dat z ascii formátu

```
v.in.ascii -n input=<vstup.asc> output=<výstup> format=standard --o
```

1.5.2.2 Převod vektorové vrstvy na rastrovou vrstvu**Převod vektorové vrstvy na Masku**

```
v.to.rast input=<vstup> output=MASK use=val --o
```

Uložením výstupní vrstvy jako MASK je zajištěno, že statistické výpočty budou prováděny pouze na území odpovídající masce. Toho lze využít například pro výpočet kubatury na zadaném území mezi dvěma vrstvami. Je-li k dispozici geometrie požadovaného území, stačí jí převést na ascii formát, načíst do GRASSu, převést na rastrovou vrstvu jako masku a použít na rozdílovou vrstvu statistické nástroje. Ze statistických výstupů lze pak jednoduše spočítat kubaturu, minimální a maximální rozdíl mezi etapami a samozřejmě výměru.

1.5.2.3 Statistické a informační funkce

K zobrazení základních informací slouží příkaz `v.info`. K jednoduchým výpočtům jako je výpočet výměry plošných objektů či délky liniových objektů slouží modul `v.report`.

Zobrazení základních informací o vektorové vrstvě

```
v.info map=<vstup>
```

Výpočet výměry plošných objektů a délky liniových objektů

```
v.report map=<vstup> option=area units=hectares  
v.report map=<vstup> option=length units=km
```

1.6 MapServer

Následující podkapitola je zpracována podle dokumentace v anglickém jazyce, uvedené na oficiálních stránkách projektu <http://mapserver.org/>.

MapServer je program, jehož cílem je dynamické zobrazování prostorových dat přes Internet či jinou síť. Projekt byl původně vyvíjen University of Minnesota¹⁴ (UMN) ve spolupráci s NASA a Minnesota Department of Natural Resources. V současné době je MapServer vyvíjen jako open-source projekt pod licencí podobnou licenci MIT a je řízen

¹⁴ Webové stránky univerzity: <http://www1.umn.edu/twincities/index.php>

*MapServer Project Steering Committee*¹⁵ (PSC) schválenou nadací *OSGeo* spravuje projekt s podporou rostoucího počtu vývojářů z celého světa.

Původně byl vyvíjen pro operační systémy UNIX a GNU/Linux, ale nyní je již MapServer multiplatformní a běží i pod jinými OS (Windows, Mac OS X, Solaris). MapServer umí s využitím knihoven GDAL a OGR pracovat s velkým množstvím rastrových a vektorových formátů. Podporuje množství OGC standardů – například WMS, WFS, WCS. Mapserver podporuje s využitím knihovny Proj.4¹⁶ více než tisíc souřadnicových systémů.

MapServer je ve své podstatě CGI program umístěn na webovém serveru ve složce se skripty, kde může být přes HTTP spuštěn. Když je na něj poslán požadavek, MapServer z parametrů URL zjistí cestu k souboru, který se nazývá *mapfile*. Na základě tohoto konfiguračního souboru a ostatních parametrů URL pak vytvoří obrázek požadované mapy. Podle požadavku může program místo obrázků mapy vrátit také například obrázek legendy, grafického měřítka, referenční mapy, nebo HTML stránku, vygenerovanou na základě šablony.

Základní součástí aplikace MapServer tvoří konfigurační soubor označovaný jako *MapFile*, geografická data a případně HTML šablony. *MapFile* je strukturovaný textový konfigurační soubor, v kterém je definován rozsah mapy, umístění dat, mapové zobrazení a jednotlivé vrstvy.

Ukázka struktury MapFile používané v aplikaci

```
MAP
  EXTENT <souřadnice LD a PH rohu mapového výřezu>
  SIZE <šířka a výška výstupního obrázku v pixelech>
  UNITS <jednotky souřadnicového systému>
  SHAPEPATH <cesta k datům>
  IMAGECOLOR <RGB barvy pozadí mapy>
  FONTSET <cesta k seznamu s fonty>
  WEB
    IMAGEPATH <systémová cesta k ukládaným obrázkům>
    IMAGEURL <URL adresa k ukládaným obrázkům>
  METADATA
    "wms_title" <název wms služby>
    "wms_onlineresource" <adresa wms služby>
    "wms_src" <souřadnicový systém>
  END
END
```

¹⁵ Směrnice PSC dostupná z WWW: <http://mapserver.org/development/rfc/ms-rfc-23.html>

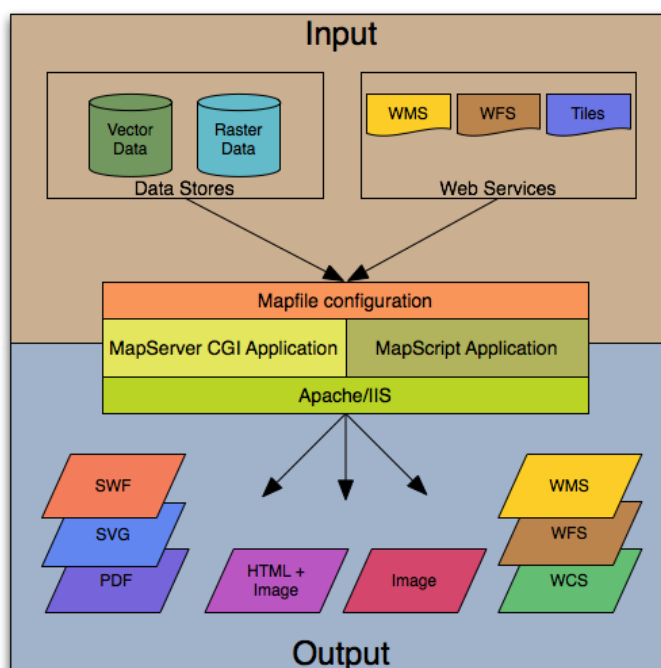
¹⁶ PROJ.4 – open-source kartografická knihovna používaná k transformaci mezi kartografickými zobrazeními. Stránky projektu: <http://trac.osgeo.org/proj/>

```
< ... definice vrstev >
END
```

Ukázka definice vrstvy wms vrstvy

```
LAYER
  NAME CUZK
  CONNECTIONTYPE WMS
  CONNECTION http://wms.cuzk.cz/wms.asp
  TYPE RASTER
  STATUS OFF
  PROJECTION
    "init=epsg:102067"
  END
  METADATA
    "wms_title" "RST_KN"
    "wms_name" "RST_KN"
    "wms_src" "EPSG:102067"
    "wms_format" "image/png"
    "wms_server_version" "1.1.0"
  END
END
```

Obrázek 6: Architektura aplikace MapServeru



Zdroj: <http://mapserver.org>

Díky využití knihoven GDAL a OGR je podporováno velké množství rastrových a vektorových formátů:

- vektorová data – např. ESRI ShapeFile, PostgreSQL-PostGIS, GML, MapInfo
- rastrová data – např. Tiff/GeoTiff, GIF, PNG, JPEG a EPPL7, GRASS, Jpeg2000

1.7 TileCache

Hlavním zdrojem informací této části kapitoly je dokumentace na oficiálních stránkách projektu <http://tilecache.org/>.

TileCache je implementace WMS-C serveru napsaného v jazyce Python, uvolněná pod BSD licenci. TileCache slouží jako server poskytující služby WMS-C a TMS. Jeho cílem je optimalizovat čas přenosu obrázků přes Internet ukládáním náhledů (dlaždic) do paměti (cache).

Pro nejjednodušší použití potřebuje aplikace ke svému chodu pouze práva číst a zapisovat na disk, spouštět CGI skripty v Pythonu a přístup k WMS, či MapServeru, k čemuž je ještě navíc potřeba Python mapscript. S těmito prostředky je možné vytvořit vlastní WMS-C server, který je možné využít s některým klientem podporujícím WMS-C nebo TMS jako například OpenLayers.

TileCache při prvním požadavku klienta (např. webová mapová aplikace) na dlaždici funguje jako prostředník mezi klientem a jiným WMS serverem – požadovanou dlaždici si vyžádá od WMS serveru a vrátí klientovi. Zároveň si však tuto dlaždici uloží do své paměti a při příštím požadavku na stejnou dlaždici již nezatěžuje WMS server, ale načte si ji přímo z paměti disku. Díky tomu je pak dosaženo rychlejší odezvy.

Na druhou stranu je třeba mít na paměti, že používání TileCache sebou přináší také řadu nevýhod: pomalejší odezva při prvním požadavku na dlaždici, dlaždice se vytváří pouze ve fixních rozlišeních, vytváří se velké množství souborů (hrozí dosažení limitu některých souborových systémů), náhledy zabírají obvykle mnohonásobně více místa než původní data. Z těchto důvodů je třeba si dobře rozmyslet, kdy tento program použít.

1.7.1 Konfigurace TileCache

Nastavení Tilecache se provádí editací konfiguračního souboru – defaultně je jím soubor `tilecache.cfg` (pod unixovými operačními systémy obvykle umístěný ve složce `/etc/`). V tomto souboru je u každé vrstvy uveden v hranatých závorkách její název, za kterým následuje sada parametrů.

Nejdůležitější parametry vrstvy

<code>bbox</code>	Souřadnice rohů ohraničujícího obdélníku vrstvy. Všechna rozlišení v sadě jsou pak defaultně nastavena tak, že se rovnají <code>bbox / 512</code> , což odpovídá velikosti dvou standardních dlaždic.
<code>extension</code>	Formát obrázků WMS
<code>layers</code>	Názvy vrstev
<code>levels</code>	Počet úrovní přiblížení
<code>mapfile</code>	Umístění <code>mapfile</code>
<code>maxResolution</code>	Maximální rozlišení. Pokud je zadáno, pole rozlišení je automaticky vypočteno podle počtu úrovní.
<code>resolutions</code>	Seznam rozlišení
<code>size</code>	Velikost generovaných dlaždic. Výchozí hodnotou je 256,256.
<code>srs</code>	Řetězec popisující souřadnicový systém. Výchozí hodnota je "EPSG:4326".
<code>Type</code>	Typ vrstvy, možnosti jsou: <code>WMSLayer</code> , <code>MapnikLayer</code> , <code>MapServerLayer</code> , <code>ImageLayer</code> .
<code>url</code>	URL adresa WMS serveru.
<code>extent_type</code>	Při nastavení na 'loose' TileCache umožní generovat dlaždice i mimo ohraničující obdélník. Vhodné pro uživatele, kteří neznají přesný rozsah.

Ukázka konfigurace WMS vrstvy

```
[Cenia]
type=WMS
url=http://geoportal.cenia.cz/wmsconnector/com.esri.wms.Esrimap/cenia_
b_ortorgb05m_sde?transparent=false
layers=0
extension=jpeg
size=256,256
bbox=-514223,-1145582,-511919,-1144256
extent_type=loose
maxResolution=2.5
levels=7
srs=EPSG:102067
```

1.7.2 Seeding

Mezi jednu z nevýhod používání TileCache patří pomalá odezva při prvním požadavku na dlaždici, která plyne z podstaty toho, že ke zdroji nepřistupujeme přímo, ale přes prostředníka. To se pak nepříjemně projeví na straně klienta, když uživatel spustí aplikaci poprvé a když v paměti nejsou uloženy žádné dlaždice. Tento problém řeší utilita `tilecache_seed.py`, která slouží k vytvoření dlaždic předtím, než je klient bude sám vyžadovat. Použití je jednoduché:

```
tilecache_seed.py <layer> [<zoom start> <zoom stop>] [options]
```

Volitelné parametry programu `tilecache_seed.py`

-f -force	Vytváří znovu již vytvořené dlaždice.
-b BBOX --bbox=BBOX	Vytváří dlaždice pouze v ohraničeném území.
-p PADDING --padding=PADDING	Pro PADDING=0 (nastaveno defaultně) se nemusí některé krajní dlaždice vytvořit. Pro PADDING=1 se vytvářejí extra krajní dlaždice, které mohou být úplně mimo ohraničené území.
-r -reverse	Vytváří dlaždice v obráceném pořadí

2. POUŽITÉ TECHNOLOGIE NA STRANĚ KLIENTA

2.1 Úvod

Webové stránky jsou navrženy tak, aby se svým vzhledem a dynamickým chováním blížili co nejvíce klasickým desktopovým aplikacím. Kromě statických HTML stránek a kaskádových stylů CSS jsou z velké části napsány v JavaScriptu, který umožňuje jejich dynamiku.

2.2 JavaScript

Programovací jazyk JavaScript je skriptovací jazyk, který se obvykle používá jako interpretovací jazyk pro webové stránky. Jeho základními vlastnostmi jsou dynamická typová kontrola, slabá typová kontrola, duck-typing (viz. Python). V JavaScriptu jsou objekty asociativní pole, jejichž klíče jsou atributy objektu, rozšířené o prototypy. Přestože je JavaScript objektově orientovaný, neobsahuje obvyklý přístup třídy a instance. Dědičnosti a jiných objektových vlastností je částečně dosaženo právě pomocí prototypů. V JavaScriptu se funkce používají zároveň jako konstruktory objektů. Voláním funkce s operátorem new dojde k vytvoření nového objektu.

Ukázka dědičnosti v programovacím jazyku JavaScript pomocí prototypů

```
function rodic() {  
    this.vlastnost_rodice = "A";  
}  
function potomek () {  
    this.vlastnost_potomka = "B"  
}  
potomek.prototype = new rodic(); # dedeni  
var o = new potomek();  
for(i in o) {alert(o[i]);}          # objekt o má obě vlastnosti
```

Následující kapitoly pojednávají o použitých javascriptových knihovnách:

- Ext JS 3.2.0
- MapFish 1.1 client
- OpenLayers 2.8

2.3 Ext JS

Ext je javascriptová knihovna pro tvorbu interaktivních webových aplikací. Tato knihovna byla původně vyvíjena jako rozšíření jiné knihovny YUI¹⁷. Od verze 1.1 se však transformovala v samostatný framework nezávislý na externích knihovnách. Framework obsahuje adaptéry (skripty) pro vzájemnou operabilitu s knihovnami YUI, jQuery¹⁸ a Prototype¹⁹.

Obrázek 7: Logo Ext JS



Zdroj: <http://www.extjs.com/assets/images/index/logo02.png>

Hlavní přínos pro programátora internetových stránek při používání této knihovny spočívá v usnadnění práce s tvorbou grafického uživatelského prostředí, vkládáním a správou událostí, s DOM a technologií AJAX. Ext dále řeší problémy vzniklé nejednotným chováním prohlížečů. Ext podporuje nejrozšířenější webové prohlížeče:

- Internet Explorer 6+
- Firefox 1.5+ (PC, Mac)
- Safari 3+
- Chrome 3+
- Opera 9+ (PC, Mac)

Zatímco jádro knihovny Ext Core je vydáváno pod svobodnou licenci typu MIT, celá knihovna Ext je vydávána pod dvěma různými typy licencí - komerční a open-source.²⁰ Komerční verze, jejíž cena se pohybuje od 729\$ pro jednotlivce²¹, umožňuje vývojáři vydávat vlastní aplikace využívající Ext pod libovolnou licenci. Zatímco vývojář používající open-source licenci (GPL) musí svůj kód opět uvolnit pod stejnou licenci.

K používání knihovny je třeba načíst tři soubory: ext-base.js, ext-all.js, ext-all.css. Přestože jsou všechny soubory minimalizovány programem YUI compressor²², jejich velikost pro verzi 3.2.0 činí cca 825kB. Ke knihovně jsou dodány lokalizační soubory

¹⁷ Yahoo! User Interface Library, oficiální stránky projektu: <http://developer.yahoo.com/yui/>

¹⁸ jQuery, malá, lehká JS knihovna, oficiální stránky projektu: <http://jquery.com/>

¹⁹ Prototype, rozsáhlejší javascriptový framework, <http://www.prototypejs.org/>

²⁰ Duální model licencí, zdroj: <http://www.extjs.com/company/dual.php>

²¹ Zdroj: <http://www.extjs.com/store/js/>, 28. 4. 2010

²² YUI compressor, program provádějící redukci javascriptového kódu odstraněním komentářů a minimalizací názvů vnitřních proměnných. www: <http://developer.yahoo.com/yui/compressor/>

obsahující překlad použitých textů do 45 jazyků včetně češtiny. Tento překlad není plně přeložen a aktualizován.

2.3.1 Práce s AJAX

AJAX (akronym anglického Asynchronous JavaScript and XML²³) je skupina vzájemně propojených technik používaných k vytvoření interaktivních webových aplikací nebo RIA. S technologií AJAX webové aplikace mohou asynchronně ze serveru načítat data na pozadí bez zasahování do vzhledu a chování stránky. Data jsou získávána pomocí objektu XHR(XMLHttpRequest). Odlišný přístup v implementaci XHR jednotlivými prohlížeči sebou přináší problémy, které knihovny pracující s technologií AJAX řeší vytvořením abstraktních prostředků, které tyto rozdíly eliminují. V knihovně Ext je tímto prostředkem singleton objekt `Ext.Ajax`.

Ukázka práce se singletonem `Ext.Ajax`

```
Ext.Ajax.request({
    url: 'prikklad.json',
    success: function(response, opts) {
        var obj = Ext.decode(response.responseText);
        console.dir(obj);
    },
    failure: function(response, opts) {
        console.log('server selhal, status code:' + response.status);
    }
});
```

2.3.2 Práce s DOM

DOM (akronym anglického Document Object Model – objektový model dokumentu) je objektově orientovaná reprezentace XML nebo HTML dokumentu. DOM je API umožňující přístup či modifikaci obsahu, struktury, nebo stylu dokumentu, či jeho částí. [16-Wikipedia]

V Ext existuje užitečná třída `Ext.DomHelper` ulehčující práci s DOM. Tato utilita slouží k dynamickému generování značek bez ohledu na použitý prohlížeč. `Ext.DomHelper` je navržen tak, aby využíval HTML fragmenty a `innerHTML` tam, kde je to výhodné pro optimalizaci rychlosti bez ztráty kompatibility. `Ext.DomHelper` je singleton a tudíž všechny jeho metody jsou staticky přístupné bez inicializace nové instance.

²³ eXtensible Markup Language, rozšiřitelný značkovací jazyk

Ukázka práce s třídou DomHelper – přidání značky div s odkazem na stránky serveru maps do těla dokumentu pomocí statické metody append:

```
Ext.DomHelper.append(document.body, {
    id: 'div',
    cn: [{
        tag: 'a',
        href: 'http://maps.fsv.cvut.cz/',
        html: 'maps',
        target: '_blank'
    }]
});
```

2.3.3 Užitečné funkce

Framework obsahuje řadu užitečných funkcí pro práci s JavaScriptem, především zpracováváním různých formátů.

2.3.3.1 Funkce Ext.apply a Ext.applyIf

Tyto funkce jsou používány pro kopírování atributů jednoho objektu do druhého objektu. První parametr je cílový objekt a druhý je zdroj atributů a jejich fixních hodnot. Funkce `Ext.applyIf` zkopíruje pouze atributy, které v cílovém objektu nejsou nadefinovány.

2.3.3.2 Url encoding/decoding

Tyto metody jsou užitečné pro transformaci javascriptového objektu obsahujícího literály do formátu, který může být poslán jako část GET řetězce, nebo obráceným směrem. `Ext.urlEncode` slouží k převedení javascriptového objektu na GET řetězec a `Ext.urlDecode` naopak k převedení GET řetězce na javascriptový objekt.

2.3.3.3 Práce s JSON

JSON (Javascript Object Notation²⁴) je datový výměnný formát, který je pro lidi lehké číst a zapisovat a pro počítače zpracovávat a generovat. Funkce `Ext.encode` a `Ext.decode` slouží k transformaci mezi javascriptovým objektem a řetězcem ve formátu JSON. Při programování je užitečná především druhá metoda, která se používá při zpracovávání odpovědi ve formátu JSON na AJAX požadavek. V navržené aplikaci jsou výstupy cgi scriptů na straně serveru posílány právě ve formátu JSON.

V navržené aplikaci jsou výstupy cgi scriptů na straně serveru posílány právě ve formátu JSON.

²⁴ popis formátu: <http://www.json.org/>

2.3.3.4 XmlReader

XML je vedle JSON další velmi používaný formát pro přenášení dat mezi klientem a serverem. K jeho zpracování slouží metody objektu `XmlReader`.

2.3.3.5 Práce s poli

Knihovna zjednodušuje práci s poli. Funkce `Ext.each` umožňuje přehlednější zápis cyklů. `Ext.toArray` převádí iterovatelný objekt nebo single objektu na pole.

2.3.3.6 Rozpoznání prohlížeče a operačního systému

Ext nabízí množství funkcí podle kterých lze rozpoznat používaný operační systém a webový prohlížeč.

2.3.3.7 Zjednodušený zápis

Pro vytváření instancí tříd je možno použít zadávání pomocí tzv. konfiguračních objektů, a pro zjednodušení zápisu do toho objektu Ext JS zavádí identifikátor typu, ke kterému je registrována příslušná třída, tzv. `xtype`. [8-Juhos]

2.3.4 GUI objekty

Knihovna obsahuje množství grafických objektů, které umožní přiblížit webovou aplikaci klasickým desktopovým aplikacím.

2.3.4.1 Tabulky a stromy

Knihovna nabízí řadu komponentů pro zobrazování velkého množství dat. Pro zobrazování tabulkových dat slouží `DataGrid` a `GridView`, pro zobrazování hierarchicky strukturovaných dat objekt `Tree`. Všechny datové komponenty lze vyplňovat manuálně nebo automatickým zpracováním dat ve formátech JSON či XML. U tabulek je možné měnit šířky sloupců, třídit a filtrovat záznamy. Mezi komponenty lze data přesouvat metodou „drag and drop“.

2.3.4.2 Formuláře

Ext implementuje klasické formulářové prvky jako `textbox`, `textarea`, `checkbox`, `radiobutton` a `combobox`, který je rozšířen o funkci našeptávače. Navíc jsou přidány prvky `spinnerfield` (rozšířený `textbox` pro editaci čísel), DHTML editor a prvek pro editaci času a data. Ke všem formulářovým prvkům je přidána možnost validace podle regulárních výrazů nebo vytvořené funkce.

Ukázka skriptu na vytvoření přihlašovacího formuláře

```
Ext.onReady(function() {
    var loginForm = new Ext.FormPanel({
        labelWidth: 75,
        frame:true,
        title: 'Login',
        bodyStyle:'padding:5px 10px 0',
        width: 300,
        autoHeight:true,
        defaults: {
            width: 230,
            anchor: '95%'
        },
        items: [{
            fieldLabel: 'Jméno',
            name: 'username',
            value: 'Adam'
        }, {
            fieldLabel: 'Heslo',
            name: 'password',
            inputType: 'password',
        }],
        buttons: [{
            text: 'Přihlásit se',
            handler: function(){
                loginForm.getForm().submit({
                    url: pyDir + 'login.py',
                    waitMsg: 'Ověřuji Vaše heslo',
                    waitTitle:'Přihlášení',
                    success: function(loginForm, o){
                        // zpracování odpovědi
                    },
                    failure: function(loginForm, o){
                        Ext.Msg.alert('Chyba', o.result.error);
                    }
                });
            }
        }
    ]
});
loginForm.render(document.body);
});
```

2.3.4.3 Panely, okna a dialogy

K uspořádání komponentů slouží objekt `Ext.Container`, který v sobě může obsahovat další komponenty, rozvržené podle layoutu. Od `Ext.Container` je odvozen objekt `Ext.Panel`, který má hlavičku, horní a dolní lištu s ovládacími prvky. Od tohoto objektu je odvozena řada dalších objektů. Například `Ext.Windows` rozšiřuje panel o možnost přesouvání po okně aplikace, změnu velikosti, minimalizaci a maximalizaci.

2.4 OpenLayers

Tato kapitola je zpracována podle dokumentace na stránkách projektu.

OpenLayers je čistě javascriptová knihovna pro vizualizaci mapových dat v nejmodernějších webových prohlížečích. OpenLayers implementuje aplikační javascriptové rozhraní pro tvorbu interaktivních geografických webových aplikací, podobných Google Maps a MSN Virtual Earth API, na rozdíl od kterých OpenLayers je svobodný software vyvíjený open-source software komunitou. Knihovna je vydávána pod licencí typu BSD.²⁵

OpenLayers implementuje protokoly standardních služeb WMS a WFS. OpenLayers je psaný v objektově orientovaném stylu užívající komponenty knihoven Prototype²⁶ a Rico²⁷.

K přidání jednoduché mapy do naší aplikace stačí do těla dokumentu umístit div značku, do které chceme vykreslovat mapu, a v javascriptu vytvořit objekt `map` konstruktorem `OpenLayers.Map` volaným s parametrem odkazujícím na tuto div značku. Do mapy je pak již potřeba jenom přidat vrstvy a ovládací prvky. Knihovna OpenLayers je navržena tak, že odděluje nástroje od mapových dat, aby všechny nástroje mohly pracovat na všech zdrojích dat.

²⁵ Text licence OpenLayers: <http://svn.openlayers.org/trunk/openlayers/license.txt>

²⁶ Prototype, rozsáhlejší javascriptový framework, <http://www.prototypejs.org/>

²⁷ Open-source javascriptová knihovna, oficiální stránky projektu: <http://openrico.org/>

Ukázka vytvoření jednoduché mapy pomocí OpenLayers

```

<html>
<head>
  <script src="cesta_k_OpenLayers/OpenLayers.js"></script>
  <script type="text/javascript">
    var map = null;
    function init(){
      map = new OpenLayers.Map('map');
      var wms = new OpenLayers.Layer.WMS(
        "OpenLayers WMS",
        "http://labs.metacarta.com/wms/vmap0?",
        {layers: 'basic'}
      );
      map.addLayer(wms);
      map.zoomToMaxExtent();
    }
  </script>
</head>
<body onload="init()">
  <div id="map">
</body>
</html>

```

2.5 MapFish Client

Framework MapFish je složen ze dvou částí: MapFish Client a MapFish Server. MapFish Server je založen na webovém frameworku napsaném v Pythonu Pylons²⁸, který dále rozšiřuje o geoprostorovou funkcionalitu.

Obrázek 8: Logo MapFish

Zdroj: <http://mapfish.org/>

MapFish Client je založen na knihovnách ExtJS a GeoExt²⁹ pro tvorbu GUI a OpenLayers pro práci s mapami. V aplikaci byla použita pouze tato část frameworku kvůli grafickým objektům a ovládacím prvkům (např. `LayerTree`, `Toolbar`).

²⁸ Pylons - framework pro tvorbu webových aplikací, oficiální stránky: <http://pylonshq.com/>

²⁹ GeoExt - JS framework založený na OpenLayers a Ext JS, www: <http://www.geoext.org/>

3. POPIS VLASTNÍ APLIKACE

3.1 Úvod

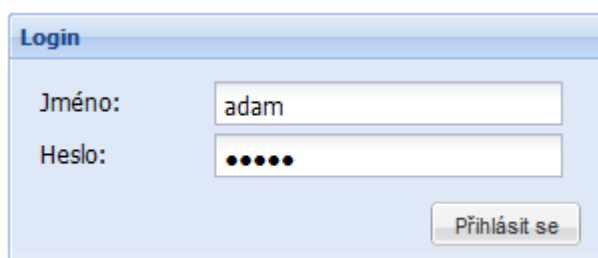
Aplikace byla primárně vyvíjena pro Mozilla Firefox 3.6.3 a dále testována na prohlížečích Internet Explorer 8.0, Google Chrome 4.1 a Opera 10.51. Na testovaných prohlížečích je aplikace funkční, liší se pouze nepatrně v grafickém provedení.

3.2 Ovládání

3.2.1 Přihlášení

Před prací s aplikací musí uživatel vyplnit uživatelské jméno a heslo a ověřit tak své oprávnění pracovat s aplikací.

Obrázek 9: Přihlašovací formulář



The image shows a web-based login form. At the top, there is a blue header bar with the word "Login" in white. Below this, the form has a light blue background. It contains two labels, "Jméno:" and "Heslo:", each followed by a text input field. The "Jméno:" field contains the text "adam". The "Heslo:" field contains seven black dots, indicating a password. To the right of the password field is a button labeled "Přihlásit se".

3.2.2 Po přihlášení

Po přihlášení se na horním panelu objeví jméno uživatele a tlačítko, kterým se lze odhlásit. První obrazovka, která se uživateli zobrazí, obsahuje v levé části tabulku s již existujícími projekty a vlevo od ní formulář na založení nového projektu. Formulář obsahuje tyto položky:

- Název
- Popis
- EPSG – kód souřadnicové systému, defaultně nastaven na 102067
- tilecache – při zaškrtnutí se bude při vizualizaci základních vrstev využívat program TileCache.

Obrázek 10: Existující projekty a formulář na přidání nového projektu

The screenshot shows a web application titled 'Control system'. At the top, it displays 'Uživatel: adam Odhlásit' and the URL 'http://www.controlsistem.cz'. The main content area is divided into two panels. The left panel, titled 'Projekty', contains a table with the following data:

ID	Název projektu		
12	Testovací Projekt	Edituj	Zobraz
13	xxxxxx	Edituj	Zobraz
11	Truhlík	Edituj	Zobraz
10	A	Edituj	Zobraz

The right panel, titled 'Nový projekt', contains a form with the following fields and controls:

- Název: [Text input field]
- Popis: [Text area]
- EPSG: [Text input field with value '102067']
- tilecache: [Checkbox]
- Buttons: 'Ulož projekt' and 'Reset'

Po uložení se nový nový projekt automaticky přidá do tabulky s existujícími projekty. Nově vzniklý projekt lze zatím pouze editovat. Projekt totiž nelze zobrazit, dokud neobsahuje aspoň jednu vrstvu.

3.2.3 Editace

V editačním režimu se v levé části obrazovky nachází strom obsahující přidané etapy, WMS vrstvy a dokumenty.

3.2.3.1 Změna názvu a popisu

Název a popis projektu lze změnit ve formuláři zobrazeném po kliknutí na položku Projekt.

3.2.3.2 Přidání nové vrstvy

Pro přidání nové vrstvy je třeba vyplnit jméno a popis a vybrat ze seznamu soubor umístěný na serveru, který odpovídá digitálnímu modelu terénu příslušné etapy. Přidáme-li první vrstvu, na horním panelu se objeví tlačítko „Zobraz“.

3.2.3.3 Editace existujících vrstev

Po kliknutí na danou vrstvu se zobrazí formulář umožňující změnu jména, popisu a pořadí dané vrstvy a smazání vrstvy z projektu.

3.2.3.4 Přidání nové WMS vrstvy

Formulář obsahuje položky název, zdroj, kód souřadnicového systému, vrstvy, formát, průhlednost a tilecache. Pokud je položka tilecache neoznačená, obrázky WMS vrstvy

budou stahovány přímo z externího WMS serveru, zatímco při označení je využívána cache na serveru.

Obrázek 11: Ukázka aplikace - editace projektu (formulář na přidání nové WMS vrstvy)

The screenshot shows the 'Control system' web application interface. The top navigation bar includes the user name 'Uživatel: adam', a link to 'Odhlasit', the project name 'Projekt: Testovací Projekt', and links to 'Zobraz' and 'Změň projekt'. The URL 'http://www.controlsystem.cz' is displayed on the right. The left sidebar contains a tree view of the project structure: 'Projekt' (expanded) with sub-items 'vrstvy' (expanded) containing 'Etapa 1 - Etapa 11', 'Etapa 2 - Etapa 22', 'Etapa 3 - Etapa 33', 'Etapa 4 - Etapa 44', and 'Přidej vrstvu'; 'wms vrstvy' containing 'WMS 1 - CUZK', 'WMS 2 - cenia', and 'Přidej WMS vrstvu' (highlighted); and 'dokumenty' containing 'bp 01.docx', 'Ahoj Svete.doc', 'Přidej dokument', and 'tilecache'. The main content area displays the 'Přidej novou WMS vrstvu' form. The form has a title 'cuzk cenia' and the following fields: 'title' (cucia), 'source' (http://geoportal.cenia.cz/wmsconnector/com.esri.wms.Esrimap/c), 'epsg' (102067), 'layers' (0), 'format' (jpeg), 'transparent' (false), and 'tilecache' (checkbox). At the bottom right of the form are 'Přidej' and 'Reset' buttons. At the bottom of the sidebar are 'Nahraj soubor' and 'Smazat projekt' buttons.

3.2.3.5 Editace existující WMS vrstvy

Všechny položky vyplňované při uložení (vyjma položky tilecache) lze dodatečně změnit po kliknutí na danou WMS vrstvu.

3.2.3.6 Přidání dokumentu

K projektu lze přidávat dokumenty týkající se projektu. Stačí vyplnit jméno a označit soubor nacházející se ve složce uživatele na serveru.

3.2.3.7 Editace dokumentu

Dodatečně je možné po kliknutí na příslušný dokument editovat jeho název, nebo dokument z projektu smazat. Kliknutím na tlačítko „Zobrazit“ se můžeme přesvědčit o obsahu dokumentu.

3.2.3.8 Nahrání souboru na server

Soubor, který chceme k projektu přidat jako dokument, je třeba umístit do složky uživatele na serveru. Jedním ze způsobů, jak to provést, je kliknout na tlačítko „Nahraj Soubor“,

které je umístěné v dolní části levého panelu v editačním módu, v zobrazeném formuláři označit umístění souboru na lokálním disku a potvrdit tlačítkem „Ulož soubor“.

3.2.3.9 Smazání projektu

V dolní části levého panelu se nachází ještě jedno tlačítko, kterým je možné projekt smazat. Před vlastním smazáním se ještě objeví potvrzovací okno.

3.2.3.10 Předvytvoření dlaždic TileCache

U všech vrstev, které jsou zobrazovány přes TileCache, může uživatel dopředu vytvořit dlaždice v požadovaných úrovních. Tato funkce poslouží zejména k urychlení prvního zobrazení, kdy nejsou vytvořeny žádné dlaždice. Proces je časově náročnější, a proto pokud aplikace nedostane ihned odpověď s informací, že dlaždice jsou dotvořeny, na další odpověď nečeká a nechá uživatele pracovat, zatímco na serveru proces dál pokračuje.

3.2.4 Vizualizace

Do prezentačního módu se lze přepnout po přidání aspoň jedné etapy. Při vizualizaci se uprostřed obrazovky nachází mapa a na okrajích jsou zasouvací panely.

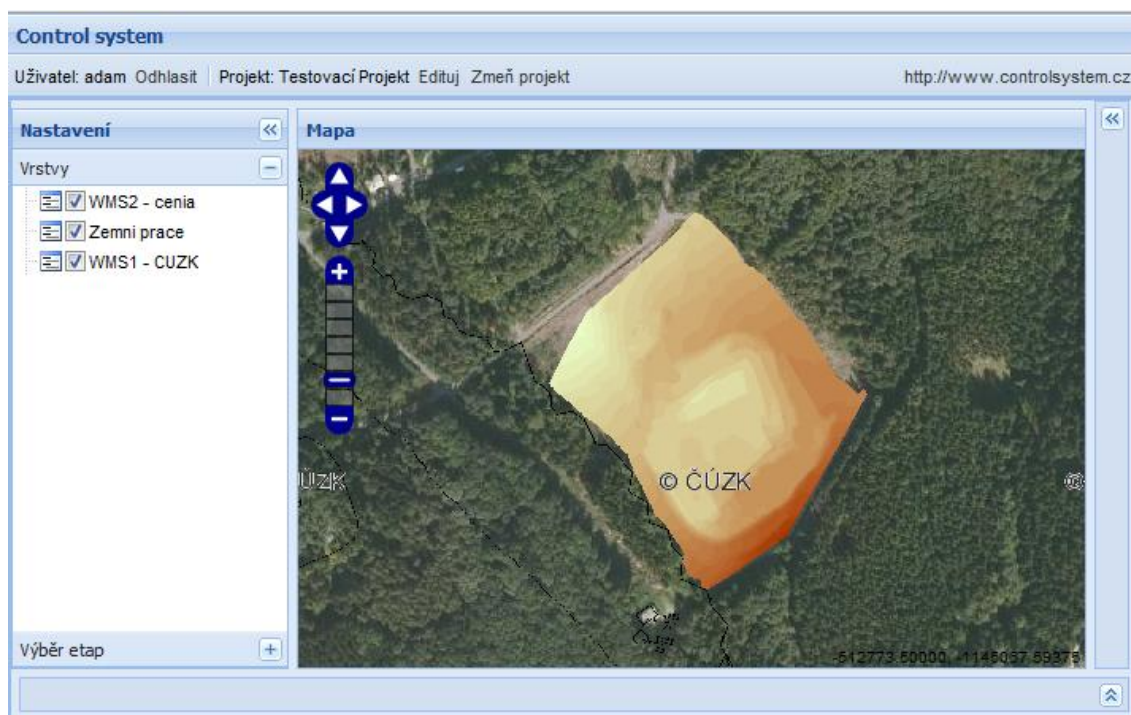
3.2.4.1 Mapa

Mapu lze ovládat pomocí standardních ovládacích prvků OpenLayers – tj. posouvat myší mapu a měnit na posouvátku v pravé části mapy měřítko ve fixních úrovních. V pravém dolním rohu jsou zobrazeny souřadnice kurzoru myši na mapě.

3.2.4.2 Nastavení

Pravý panel je určen k nastavení a obsahuje dvě záložky. Pod první záložkou „Vrstvy“ je zobrazen seznam vrstev. Seznam obsahuje hlavní vrstvu označenou „Zemni prace“ a WMS vrstvy. V seznamu je možné měnit nastavení vrstev, které se mají zobrazit, a táhnutím myši pořadí, v jakém se mají vrstvy zobrazit. Druhá záložka umožňuje měnit pomocí dvou posouvátek etapu. Pokud jsou obě posouvátka nastavena na stejnou etapu, zobrazí se výškopisný model dané etapy. Naopak jsou-li nastavena na odlišnou etapu, bude se jako hlavní vrstva zobrazovat model výškového rozdílu obou etap. Kromě dvou posouvátek na výběr etapy, je zde ještě umístěno třetí posouvátko určující viditelnost – respektive průhlednost hlavní vrstvy.

Obrázek 12: Ukázka aplikace - vizualizace DMT s WMS vrstvami



3.2.4.3 Informační panel

Dolní panel slouží jako informační. Obsahuje popis projektu, přiložené dokumenty, které si lze prohlížet či stáhnout, a popis zobrazovaných etap.

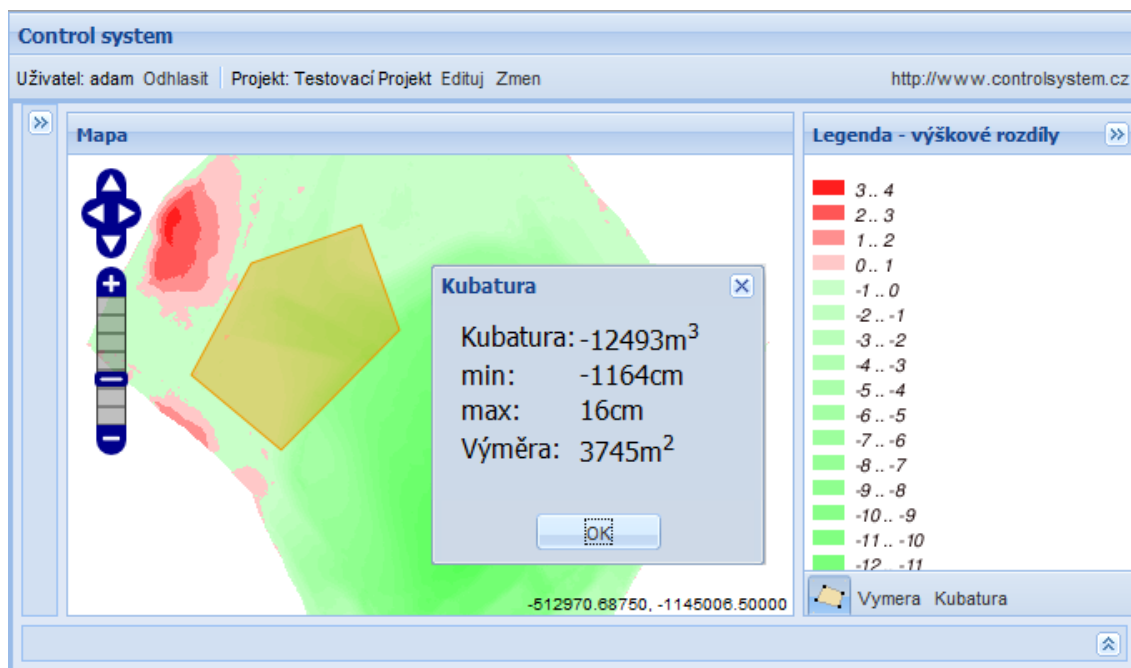
3.2.4.4 Legenda

V levé části je umístěn zasouvací panel s legendou. Legenda se automaticky mění podle toho, je-li zobrazován výškopis jedné etap, nebo výškopisný rozdíl dvou etap.

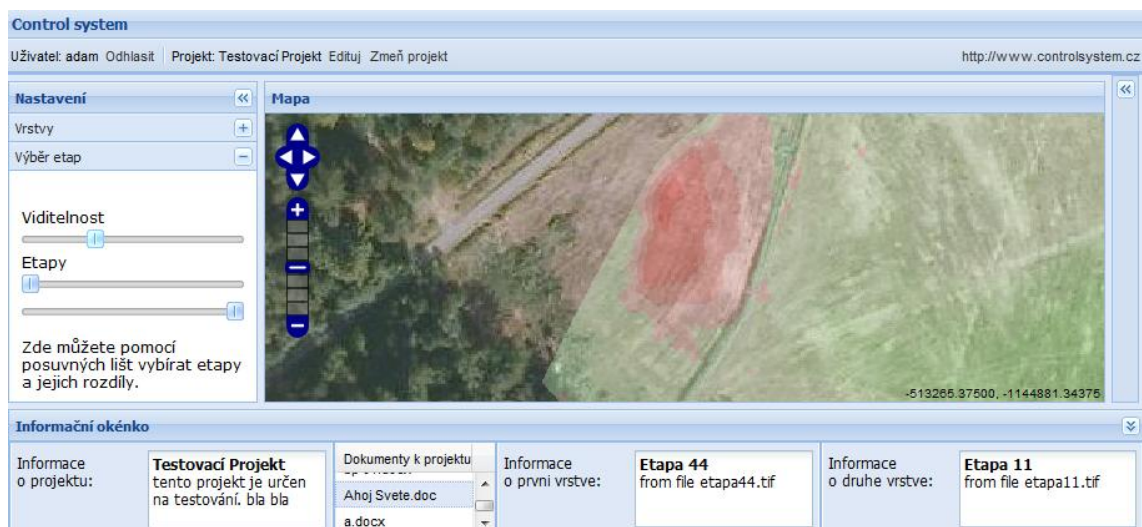
3.2.4.5 Výpočet kubatury

Při zobrazování výškopisného rozdílů dvou etap se v dolní části panelu s legendou objeví nástrojová lišta s třemi tlačítky. Po aktivaci prvního tlačítka můžeme na mapě klikáním myši označit oblast, nad kterou chceme výpočet provádět. Uzavření oblasti se provádí dvojklikem na poslední bod. Po stisknutí druhého tlačítka se vypočte výměra zadané oblasti. Třetím tlačítkem se vypočte kubatura a maximální rozsah výškového rozdílu označených etap. Výpočet kubatury probíhá na straně serveru, a proto trvá vyřízení požadavku déle.

Obrázek 13: Ukázka aplikace - výpočet kubatury



Obrázek 14: Ukázka aplikace - volba etapy a viditelnosti



ZÁVĚR

V souladu s cílem této bakalářské práce jsem navrhl a vytvořil interaktivní webový systém na správu mapových projektů.

Webová aplikace je dostupná k prohlídnutí na stránce:

<http://maps.fsv.cvut.cz/~polivada/controlsystem/>

Klientská část systému je postavena na frameworku Ext JS, který využívá nejmodernější techniky webových aplikací – asynchronní komunikaci se serverem, vizuální efekty, kaskádové styly, grafické komponenty. Aplikace je tudíž vizuálně přívětivá a snadno ovladatelná.

Při vizualizaci bylo třeba řešit problém s nízkou rychlostí zobrazování vrstev, který je způsoben zasláním velkého počtu požadavků na MapServer v jeden časový okamžik. Tento problém řešil software TileCache, který na straně serveru dopředu ukládá výsledné obrázky, čímž ušetří čas procesoru potřebný k odesílání obrázků klientovi. Uživatel může nastavit používání TileCache, jak na základní vrstvy DMT, tak na připojené WMS vrstvy. Vzhledem k tomu, že načítání WMS služeb zaměstnává jiný server, je pouze na uživateli, aby zhodnotil, u kterých vrstev TileCache používat.

Aplikace obsahuje všechny funkce, které byly uvedeny v zadání. Při vizualizaci navíc, kromě sledování změn výškových modelů, aplikace nabízí funkci na výpočet plochy a kubatury v uživatelem označeném území.

Přínos této práce shledávám především v tom, že výsledkem je produkt, který bude s největší pravděpodobností používán v praxi.

POUŽITÉ ZDROJE INFORMACÍ

- [1] **OpenLayers: Home.** [Online] [Citace: 25. Duben 2010]
Dostupný z WWW: <<http://openlayers.org/>>.
- [2] **Camptocamp. MapFish.** [Online] [Citace: 24. 4 2010]
Dostupný z WWW: <<http://mapfish.org/>>.
- [3] **Ext JS, Inc. Ext Core Manual.** [Online] [Citace: 25. Duben 2010]
Dostupný z WWW:<<http://www.extjs.com/products/core/manual/>>.
- [4] **Ext JS, Inc. Ext - Licensing Overview.** [Online] [Citace: 25. Duben 2010]
Dostupný z WWW:<<http://www.extjs.com/products/license.php>>.
- [5] **Garrett, Jesse James. 2008.** Ajax: A New Approach to Web Applications. *Adaptive Path.* [Online] 18. Únor 2008. [Citace: 19. Duben 2010] Dostupný z WWW:<<http://www.adaptivepath.com/ideas/essays/archives/000385.php>>.
- [6] **GRASS Development Team. Vector data processing in GRASS GIS.** [Online] 2008. [Citace: 25. Duben 2010] Dostupný z WWW:
<http://grass.osgeo.org/grass64/manuals/html64_user/vectorintro.html>.
- [7] **GRASS Development Team. Raster data processing in GRASS GIS.** [Online] 2008. [Citace: 25. Duben 2010] Dostupný z WWW:
<http://grass.osgeo.org/grass64/manuals/html64_user/rasterintro.html>.
- [8] **Juhos, Miroslav.** Ext JS - javascriptový framework pro tvorbu RIA - Zdroják. *root.cz.* [Online] [Citace: 20. 4 2010] Dostupný z WWW:
<<http://zdrojak.root.cz/clanky/ext-js-javascriptovy-framework-pro-tvorbu-ria>>
- [9] **MetaCarta. TileCache -- Web Map Tile Caching.** [Online] [Citace: 25. Duben 2010]
Dostupný z WWW: <<http://tilecache.org/>>.
- [10] **Regents of the University of Minnesota. Mapserver 5.6.1 documentation.** [Online] [Citace: 25. Duben 2010] Dostupný z WWW:<<http://mapserver.org/>>.
- [11] **Regents of the University of Minnesota. MapServer 5.6.1 Documentation.** [Online] [Citace: 25. Duben 2010] Dostupný z WWW:
<<http://mapserver.org/documentation.html#documentation>>.
- [12] **Stěhule, Pavel. PostgreSQL.** [Online] [Citace: 20. Duben 2010]
Dostupný z WWW: <<http://www.postgres.cz/index.php/PostgreSQL>>.
- [13] **The Apache Software Foundation.** The Apache HTTP Server Project. [Online] [Citace: 25. Duben 2010] Dostupný z WWW:<<http://httpd.apache.org/>>.
- [14] **The PostgreSQL Global Development Group. PostgreSQL.** [Online] [Citace: 25. Duben 2010] Dostupný z WWW:<<http://www.postgresql.org/>>.
- [15] **The PostgreSQL Global Development Group.** PostgreSQL 8.1.20 Documentation. [Online] [Citace: 25. Duben 2010]
Dostupný z WWW:<<http://www.postgresql.org/docs/8.1/static/index.html>>.

- [16] **Wikipedie, Otevřená encyklopedie.** Document Object Model. [Online] 13. Prosinec 2009. [Citace: 27. 4. 2010] Dostupný z WWW: <http://cs.wikipedia.org/w/index.php?title=Document_Object_Model&oldid=4705434>.
- [17] **Wikipedia, Free Encyklopedia.** Ext (JavaScript library). [Online] 4. Březen 2010. [Citace: 8. Duben 2010] Dostupný z WWW: <[http://en.wikipedia.org/w/index.php?title=Ext_\(JavaScript_library\)&oldid=359996535](http://en.wikipedia.org/w/index.php?title=Ext_(JavaScript_library)&oldid=359996535)>.
- [18] **Wikipedia, Free Encyklopedia.** JavaScript. [Online] 7. Březen 2010. [Citace: 20. Duben 2010] Dostupný z WWW: <<http://en.wikipedia.org/w/index.php?title=JavaScript&oldid=360639810>>.
- [19] **Wikipedia, Free Encyklopedia.** Python (programming language). [Online] 5. Březen 2010. [Citace: 19. Duben 2010] Dostupný z WWW: <[http://en.wikipedia.org/w/index.php?title=Python_\(programming_language\)&oldid=360381440](http://en.wikipedia.org/w/index.php?title=Python_(programming_language)&oldid=360381440)>.

POUŽITÉ ZKRATKY A TERMÍNY

AJAX	Asynchronous JavaScript and XML
CGI	Common Gateway Interface
CSS	Cascading Style Sheet
DHTML	Dynamic HTML
DOM	Document Object Model – objektový model dokumentu
GDAL	Geospatial Data Abstraction Library - rastrová knihovna, oficiální stránky: http://www.gdal.org/
GeoTIFF	Geographic Tagged Image File Format
GIS	Geografický informační systém
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
OGC	Open Geospatial Consortium – standardizační organizace pro geoprostorová data a služby, oficiální stránky: http://www.opengeospatial.org/
OGR	OGR Simple Features Library – C++ open-source knihovna pro práci s vektorovými daty, součást knihovny GDAL, stránky: http://www.gdal.org/ogr/
OSGeo	Open Source Geospatial Foundation, nadace podporující open-source projekty v oblasti geoinformatiky, oficiální stránky: http://www.osgeo.org/
RIA	Rich Internet Application, interaktivní webová aplikace
TMS	Tile Map Service, specifikace OSGeo: http://wiki.osgeo.org/wiki/Tile_Map_Service_Specification
UMN	University of Minnesota
URL	Uniform Resource Locator
WCS	Web Coverage Service, standard OGC, specifikace: http://www.opengeospatial.org/standards/wcs
WFS	Web Feature Service, standard OGC, specifikace: http://www.opengeospatial.org/standards/wfs
WMS	Web Mapping Service, standard OGC, specifikace: http://www.opengeospatial.org/standards/wms
WMS-C	Web Mapping Service – Cached
XML	eXtensible Markup Language – rozšiřitelný značkový jazyk
WWW	World Wide Web

SEZNAM OBRÁZKŮ

Obrázek 1: Architektura serveru	2
Obrázek 2: Základní schéma databáze	3
Obrázek 3: Logo PostgreSQL.....	4
Obrázek 4: Logo projektu Python.....	4
Obrázek 5: Logo GRASS GIS.....	8
Obrázek 6: Architektura aplikace MapServeru	14
Obrázek 7: Logo Ext JS	19
Obrázek 8: Logo MapFish	25
Obrázek 9: Přihlašovací formulář	26
Obrázek 10: Existující projekty a formulář na přidání nového projektu.....	27
Obrázek 11: Ukázka aplikace - editace projektu (formulář na přidání nové WMS vrstvy)	28
Obrázek 12: Ukázka aplikace - vizualizace DMT s WMS vrstvami	30
Obrázek 13: Ukázka aplikace - výpočet kubatury	31
Obrázek 14: Ukázka aplikace - volba etapy a viditelnosti	31