

Univerzita Palackého v Olomouci

Přírodovědecká fakulta

Katedra geoinformatiky

Markéta SOLANSKÁ

**TVORBA MODULU PRO VSTUP
A EDITACI DAT PRO MAPSERVER**

Bakalářská práce

Vedoucí práce: RNDr. Vilém Pechanec, Ph.D.

Olomouc 2012

Čestné prohlášení

Prohlašuji, že jsem bakalářskou práci bakalářského studia oboru Geoinformatika a geografie vypracovala samostatně pod vedením RNDr. Viléma Pechance, Ph.D.

Všechny použité materiály a zdroje jsou citovány s ohledem na vědeckou etiku, autorská práva a zákony na ochranu duševního vlastnictví.

Všechna poskytnutá i vytvořená digitální data nebudu bez souhlasu školy poskytovat.

V Olomouci 13. srpna 2012

Děkuji vedoucímu práce RNDr. Vilému Pechancovi, Ph.D. za podněty a připomínky při vypracování práce.

Vložený originál **zadání** bakalářské/magisterské práce (s podpisy vedoucího katedry, vedoucího práce a razítkem katedry). Ve druhém výtisku práce je vevázána fotokopie zadání.

1.strana

Vložený originál **zadání** bakalářské/magisterské práce (s podpisy vedoucího katedry, vedoucího práce a razítkem katedry). Ve druhém výtisku práce je vevázána fotokopie zadání.

2.strana

OBSAH

SEZNAM POUŽITÝCH ZKRATEK.....	8
ÚVOD.....	9
1 CÍLE PRÁCE.....	10
2 POUŽITÉ METODY A POSTUPY ZPRACOVÁNÍ.....	11
2.1 Mapové servery.....	11
2.2 Použité prostředky (programy).....	11
2.2.1 UMN MapServer.....	12
2.2.2 Apache HTTP server.....	12
2.2.3 PostgreSQL.....	13
2.2.4 PostGIS.....	13
2.2.5 PHP.....	13
2.2.6 JavaScript.....	14
2.2.7 OpenLayers.....	14
2.3 Standardy OGC.....	14
2.3.1 Web Map Service (WMS).....	14
2.3.2 Simply Feature SQL.....	15
2.4 Postup zpracování.....	16
3 SOUČASNÝ STAV ŘEŠENÉ PROBLEMATIKY.....	18
3.1 Nekomerční řešení.....	18
3.1.1 p.mapper.....	18
3.1.2 GeoMoose.....	19
3.1.3 Další nekomerční aplikace.....	19
3.2 Komerční řešení.....	20
3.2.1 T-MAPY spol. s r.o.....	20
3.2.2 Help Service – Remote Sensing spol. s r.o.....	20
3.2.3 MapSherpa.....	20
3.3 Diplomové práce.....	20
4 VÝSLEDKY.....	22
4.1 Programové komponenty.....	22
4.2 Struktura modulu.....	22
4.3 Struktura databáze.....	24
4.4 Tvorba modulu.....	24
4.4.1 Vytvoření databáze.....	24
4.4.2 PHP skript.....	26
4.4.3 Struktura MapServer template.....	26
4.4.4 Skript OpenLayers.....	27
4.5 Konfigurace.....	30

4.5.1 Nastavení absolutních cest.....	30
4.5.2 Nastavení souboru Mapfile.....	31
4.6 Popis fungování modulu.....	33
5 DISKUZE.....	37
6 ZÁVĚR.....	39
POUŽITÁ LITERATURA A INFORMAČNÍ ZDROJE	
SUMMARY	
PŘÍLOHY	

SEZNAM POUŽITÝCH ZKRATEK

Zkratka	Význam
API	Application Programming Interface
CSS	Cascading Style Sheets
CGI	Common Gateway Interface
FOSS	Free Open Source Software
GDAL	Geospatial Data Abstraction Library
GEOS	Geometry Engine Open Source
GML	Geography Markup Language
GNU	GNU is not Unix
GPX	GPS eXchange Format
HTML	Hyper Text Markup Language
HTTP	Hypertext Transfer Protokol
ESRI	Environmental System Research Institute
GIS	Geoinformatický informační systém
GIT	Geoinformační technologie
JPEG	Joint Picture Experts Group
KML	Keyhole Markup Language
PHP	Hypertext Preprocessor (původně: Personal Home Page)
PNG	Portable Network Graphics
TIFF	Tagged Image File Format
OGC	Open Geospatial Consortium
OSGEO	Open Source Geospatial Foundation
SFS	Simply Features SQL
SHP	Shapefile
SQL	Structured Query Language
URL	Uniform Resource Locator
WFS	Web Feature Service
WMS	Web Map Service
WKT	Well-Known Text
WWW	World Wide Web

ÚVOD

Informační technologie zaznamenaly v posledních letech nebývalý rozvoj. Dnes už téměř každý má přístup k počítači a umí jej ovládat. Samozřejmostí je už i internet, ke kterému díky bezdrátovým sítím není problém se připojit například i v restauraci, na náměstí či v parku. Vše směřuje ke sdílení dat v rámci počítačové sítě, či po internetu. V obou případech se nabízí, jako ideální řešení, sdílení pomocí internetového prohlížeče, kdy stránka, která se zobrazuje, je uložena na serveru. Tam jsou také uložena data, se kterými se pracuje, a odehrávají se zde i veškeré operace. To zajišťuje relativní nenáročnost na výkonnost počítače uživatele, či velikost jeho disku.

Od kreslení map pomocí tužky a pravítka se přešlo ke „klikání“ myši na obrazovce a s trochou cviku se to naučí i běžný uživatel. Používání mapových aplikací na webu se rozšiřuje natolik, že se začíná objevovat stále více uživatelů, kteří si tvoří vlastní mapy, popř. kteří přispívají při tvorbě dat pomocí projektů jako je například OpenStreetMap (viz www.openstreetmap.org). U takových projektů není vyžadována znalost kartografie a přesto uživatelé vytvářejí kvalitní data. Dnešní vývoj stále více směřuje k usnadnění přístupu k datům, což je náplní i této práce.

Jedním z požadavků této práce bylo vytvořit modul pomocí nekomerčního řešení, tedy pomocí programů, které jsou nejen zdarma stažitelné, ale většinou také snadno rozšířitelné díky tzv. otevřenému zdrojovému kódu, anglický název pro tento druh softwaru je Free Open Source Software (FOSS). Myšlenka otevřeného software vznikla v roce 1984, kdy Richard Stallman vytvořil unixový operační systém založený na této myšlence. Na základě toho pak byla formulována licence (GNU General Public License), která určuje základní zásady svobodného software.

V posledních letech se dost rozvíjí i svobodný software, určený ke zpracování a prezentování geografických dat. V této práci byl použit především software pro tvorbu a správu internetových mapových aplikací, konkrétně knihovny OpenLayers, GDAL (Geospatial Data Abstraction Library), GEOS (Geometry Engine Open Source), PROJ4, databáze PostgreSQL s rozšířením PostGIS a v neposlední řadě mapový server UMN MapServer. Přehled open source programů například na webové stránce <http://www.opensourcegis.org/>.

Mapových serverů je celá řada od těch komerčních jako je ArcGIS for Server (ESRI), Autodesk MapGuide Enterprise, GeoMedia (Intergraph) až po nekomerční UMN MapServer, GeoServer nebo Deegree.

1 CÍLE PRÁCE

Cílem bakalářské práce je navrhnout a následně vytvořit modul pro vstup a editaci vektorových dat v prostředí UMN MapServer. Nástroje budou připraveny pro webové rozhraní tenkých klientů a data se budou ukládat dle standardů OGC (Open Geospatial Consortium) do prostředí PostGIS. Nástroje budou pracovat s geometrickou a atributovou složkou geoprůvku. Na základě literární rešerše bude zpracován koncept projektové dokumentace, na jejímž základě se modul zrealizuje. Při realizaci budou respektována pravidla pro tvorbu webových aplikací a výsledný produkt bude zcela založen na nekomerčním řešení. Funkční řešení bude dále implementováno do systému ArchonIS. Teoretická část práce se bude věnovat problematice vstupu a editace prostorových geodat skrze webová rozhraní. Proces praktické části bude dostatečným způsobem zdokumentován a optimalizován.

2 POUŽITÉ METODY A POSTUPY ZPRACOVÁNÍ

V následující kapitole jsou podrobně rozebrány technologie, které byly použity pro tvorbu modulu. Dále jsou zde vysvětleny OGC standardy a detailně popsány fáze tvorby modulu.

2.1 Mapové servery

Pro zobrazení geografických dat na webu jsou nejčastěji používány mapové servery. Pracují na bázi klient-server a úzce spolupracují s webovými servery. Jako výsledek jsou zobrazovány statické mapy, nejčastěji ve formátu obrázku (př. png), nebo výsledky dotazu. Mapové servery umožňují pouze základní manipulaci s mapou, prostřednictvím konfiguračního souboru lze určit měřítko, zobrazené oblasti, definovat viditelné vrstvy, zobrazení popisu či nastavení symbologie. V kombinaci s dalšími skriptovacími jazyky umožňují interaktivní ovládání mapové aplikace, tedy změny měřítka, posouvání výřezu, změnu viditelných vrstev, identifikaci prvku či dotazování.

Fungování mapového serveru podrobně popisuje Šrejber (2002). Podle něj mapový server pracuje s geografickými daty a s atributovou složkou. Na základě požadavků klienta server zpracuje příslušná mapová a databázová data a výslednou stránku prostřednictvím WWW (World Wide Web) serveru odešle klientovi (Šrejber 2002).

Zobrazení požadované mapy v okně webového prohlížeče probíhá, podle Šrejbera (2002), ve třech krocích, které používají http protokol (Hypertext Transfer Protokol):

- 1) Webový prohlížeč (klient) zašle dotaz WWW serveru, kde je vystavena požadovaná aplikace (HTML stránka s URL adresou, obsahující např. volání CGI (Common Gateway Interface) skriptu a mapu v rastrovém či vektorovém formátu).

- 2) WWW server pošle požadavek na vyvolání CGI skriptu aplikačnímu mapovému serveru.

- 3) Mapový server CGI skript spustí a výsledek odešle WWW serveru. Ten jej doplní do HTML (Hypertext Markup Language) dokumentu, který se objeví v prohlížeči uživatele (Šrejber 2002).

V mapovém serveru lze definovat také šablony, které jsou většinou formátu HTML, nebo někdy samotné url. Používají se převážně na zobrazení výsledků prostorových dotazů. Jako výsledek lze vypsát názvy atributů a jejich hodnoty, popřípadě lze šablonu upravit jakkoliv podle možností HTML.

2.2 Použité prostředky (programy)

Modul byl vyvíjen na operačním systému GNU/Linux Mint 12 Lisa a programy byly nainstalovány z oficiálního repozitáře Ubuntu 11.10 Oneiric Ocelot a z přídatného repozitáře UbuntuGIS stable (k nalezení na adrese <https://launchpad.net/~ubuntugis/+archive/ppa/>).

2.2.1 UMN MapServer

Jako mapový server byl použit UMN MapServer, verze 6.0.3, který slouží k prezentaci prostorových dat ve formě obrazových dat pro uživatele či ve formě odpovědi na dotazy vyšší vrstvy.

Byl vyvinut v 90. letech na University of Minnesota a v dnešní době je projektem OSGEO (Open Source Geospatial Foundation), který podporuje vývoj open source aplikací zpracovávajících geoprostorová data. Na rozdíl od mnoha komerčních systémů, běží na většině operačních systémů včetně GNU/Linux.

MapServer má podporu pro skriptovací jazyky PHP, Python, Java, Perl, Ruby a .NET, dále podporuje vstup rastrových (TIFF) a vektorových (ESRI shapefile) dat, stejně tak jako dat z databází PostgreSQL (s nastavbou PostGIS), Oracle (s nastavbou Oracle Spatial), ESRI ArcSDE, MySQL a dalších formátů prostřednictvím knihovny OGR. Kompletní přehled podporovaných formátů knihovny OGR na webových stránkách <http://www.gdal.org/ogr/>.

Podpora geografických projekcí je zajištěna knihovnou PROJ4. MapServer lze použít jako CGI aplikaci nebo přes MapScript, který podporuje několik programovacích jazyků (MapServer, 2011).

Pro konfiguraci mapového pole je použit Mapfile, který obsahuje základní nastavení mapy, tedy umístění dat, určení souřadnicových systémů, symbologie, legendy, vzhledu mapového pole a dalších.

Další informace je možno si přečíst na internetové adrese <http://mapserver.org/> nebo na stránkách projektu OSGEO <http://www.osgeo.org/>.

2.2.2 Apache HTTP server

Jedná se o webový server s otevřeným kódem, který vznikl na University of Illinois už na začátku 90. let, a dnes se o jeho vývoj stará Apache software foundation. Tento open source, fungující na všech základních platformách, zpracovává http požadavky klienta a vrací mu zpět výsledky přes http protokol do prohlížeče. Může se jednat o statický obsah nebo se může výstup vytvořit dynamicky, například pomocí PHP (The Apache Software Foundation, 2012).

Dnes patří mezi nejrozšířenější webové servery a je používán po celém světě. Důvodem je například přizpůsobitelnost pomocí API (Application Programming Interface) modulů Apache, podpora dynamicky sdílených objektů, což umožňuje načítání modulů za běhu, programovatelnost pomocí různých jazyků a integračních technik či možnost zabezpečení. To je zajištěno podporou databází pro ověřování uživatelů a podporou SSL (Secure Sockets Layer), která umožňuje výměnu digitálních certifikátů a šifrování dat procházejících internetem. Apache také nabízí plné dodržování standardu HTTP/1.1 (RFC 2616), který zajišťuje podporu virtuálních hostitelů, trvalá spojení, přenášení od klienta na server, rozšíření hlášení chyb či ukládání prostředků do paměti cache (Aulds, 2003).

V průběhu práce na bakalářské práci byla používána verze 2.2.20. Více informací na stránkách <http://httpd.apache.org/>.

2.2.3 PostgreSQL

PostgreSQL je objektově-relační databázový systém s otevřeným zdrojovým kódem, fungující na všech platformách. Byl vyvinut na Univerzity of California v Berkley. PostgreSQL je jednou z nejpokročilejších databází, která je schopná pracovat s velkými objemy dat a díky své rychlosti a funkcionalitě může soupeřit i s populárními komerčními systémy jako je Oracle, IBM DB2, Microsoft SQL Server 2008 a další (PostgreSQL, 2012).

Je použita verze 9.1.4 a podrobnosti o tomto databázovém systému je možno nalézt na <http://www.postgresql.org/>.

2.2.4 PostGIS

PostGIS je rozšířením databáze PostgreSQL o podporu geografických dat. PostGIS implementuje specifikaci „Simple Features for SQL“ konsorcia OGC.

PostGIS umožňuje ukládání geometrických objektů (bod, linie, polygon), prostorové funkce pro určení vzdáleností, délky linií, výměr a obvodu ploch, výběr indexu při spojení prostorových a atributových dotazů a další. Existuje také další nastavba, která rozšiřuje ukládání a manipulaci s rastrovými daty (PostGIS Raster), nastavba pro topologickou správu vektorových dat (PostGIS Topology) a pro síťové analýzy (pgRouting). PostGIS je podporován velkou řadou softwarů zabývajících se úpravou geografických dat, což také umožňuje snadnou přenositelnost a použitelnost nastavby (příklad software s PostGIS: QGIS, GvSIG, GRASS).

PostGIS používá mnoho knihoven jako GEOS (Geometry Engine Open Source) pro implementaci jednoduchých prostorových prvků a metod pro topologii, PROJ4 pro převod mezi kartografickými projekcemi, GDAL/OGR (Geospatial Data Abstraction Library) pro převod mezi různými vektorovými i rastrovými formáty (Obe, Hsu 2011).

Byla použita verze 1.5.3, více informací na oficiálních webových stránkách <http://postgis.refractions.net/>.

2.2.5 PHP

PHP (Hypertext Preprocessor) je skriptovací dynamicky typovaný jazyk pracující na straně serveru nezávislý na platformě. Překlad PHP stránek je možné zajistit modulem Apache nebo CGI skriptem a častým použitím je kombinace s HTML kódem nebo dalšími skriptovacími jazyky jako je JavaScript, Ajax, Python. Skripty jsou zpracovávány na straně serveru a poté jsou zobrazovány klientovi pomocí webového prohlížeče.

Z hlavních výhod je právě specializovanost na tvorbu webových stránek. Díky své popularitě je možno nalézt mnoho návodů a hotových kódů přímo na internetu, má také podporu na hostingových službách. Obrovskou výhodou je nezávislost na licenci, čili na svobodné licenci bez copyleftu. Podporuje databáze, není náročný na software ani hardware klienta, protože výstupem je čisté HTML, s tím souvisí i menší objem přenesených dat a vyšší ochrana zdrojových textů programů (Brázda, 2005).

Programovací jazyk PHP je v modulu použit převážně pro komunikaci s databází, tedy zajišťuje prostředníka mezi html formulářem na straně klienta a databází na straně serveru.

Byla použita verze 5.3.6. Více informací na oficiálních stránkách <http://www.php.net/>.

2.2.6 JavaScript

JavaScript je, podobně jako PHP, objektově orientovaný skriptovací jazyk nezávislý na platformě, ale s tím rozdílem, že JavaScript pracuje na straně klienta a proto zaručuje rychlejší odezvy a větší možnosti grafického nastavení. Používá se převážně pro tvorbu dynamických stránek s interaktivními prvky jako jsou tlačítka, textová pole, animace či jiné efekty. Je často vkládán přímo do HTML kódu (Janovský, 2012).

Více informací na oficiálních stránkách <http://www.javascriptsource.com/>.

2.2.7 OpenLayers

Knihovna OpenLayers je JavaScript API určená pro tvorbu webových dynamických mapových aplikací pracujících na straně klienta. Je součástí projektu OSGEO, což mimo jiné zajišťuje dodržování OGC standardů popsaných v následující kapitole.

Byla vyvinuta za účelem kombinovat mapy z různých zdrojů, aby nebylo potřeba se zabývat rozdíly v API jednotlivých poskytovatelů. Mapové podklady, založené na systému dlaždic, poskytují například GoogleMaps, Yahoo! Maps, Bing Maps či OpenStreet Map. Také dovozuje přidat mapu ve formě obrázku postavenou na OGC standardu WMS (Web Map Services) nebo vektorovou vrstvu přes WFS (Web Feature Services) (OpenGeo, 2012).

Byla použita verze OpenLayers 2.11.

2.3 Standardy OGC

Sdílení dat mezi dvěma aplikacemi nebo prostřednictvím internetu s největší přesností zajišťují standardy pro ukládání, sdílení a výměnu dat (Mitchell, 2005). Standardy OGC jsou vyvíjené konsorciem Open Geospatial Consortium, kterou tvoří a podporuje přes 400 firem, univerzit, vládních či vědeckých organizací. Výsledkem jsou standardy, které umožňují vývojářům zpřístupnit všechny geoprostorové informace, data a služby uživatelům prostřednictvím počítačových sítí či na webu v různých typech aplikací. Umožňují také vzájemnou spolupráci při řešení geoprostorových úloh (OGC, 2012).

V této práci byly použity dva z těchto standardů, Web Map Service pro přenos dat a Simple Feature SQL (Structured Query Language) pro ukládání dat do databáze.

2.3.1 Web Map Service (WMS)

Web Map Service, česky webová mapová služba, poskytuje data ve formě rastrových map, nejčastěji ve formátech png, jpeg, gif, někdy také jako vektorová vrstva ve formátu SVG (Scalable Vector Graphics) nebo WebCGM (Web Computer Graphics Metafile).

Zdrojem těchto dat může být ESRI Shapefile, GeoTIFF, nebo data z databáze. Nastavení WMS je definováno v Mapfile (Beaujardiére, 2006).

Podle Jiráňka (2008) komunikace mezi klientem a mapovým serverem probíhá prostřednictvím třech základních dotazů. (Názvy těchto dotazů jsou platné až od verze WMS 1.1.0.):

GetMap – tento dotaz zpřístupní mapu ve formě obrazových dat v určitém formátu. Query URL musí obsahovat parametr REQUEST=GetMap. Specifikace (záleží dle použité verze WMS) vyžaduje ještě další povinné parametry, které mapovému serveru předávají podrobnější informace o zobrazované mapě. Jsou jimi například souřadnicový systém, formát obrazových dat, název vrstev a stylů, souřadnicový box, aj.

GetCapabilities – použití tohoto dotazu se většinou zvolí v případě, že klient nezná možnosti a vlastnosti spravovaných dat a musí tyto možnosti zjistit. Proto klient při první komunikaci se serverem sestaví GetCapabilities dotaz. Dotaz musí obsahovat parametry REQUEST=GetCapabilities a SERVICE=WMS. Po platném sestavení dotazu je zpřístupněn XML soubor popisující danou službu. Tyto data jsou metadaty mapového serveru tzn., že obsahují veškeré potřebné informace pro komunikaci se službou.

GetFeatureInfo – tento typ dotazu vrací klientovy XML soubor s atributy daného prvku na mapě o určitých souřadnicích. Dotaz URL musí obsahovat parametr REQUEST=GetFeatureInfo (Jiráňek, 2008).

2.3.2 Simply Feature SQL

Standard pro geografické informace Simple feature access se skládá ze dvou částí. První část Common architecture popisuje obecnou architekturu, druhá část nese název SQL option. Popisuje schéma dotazování, ukládání, vyhledávání tříd či kolekcí geoprostorových prvků s jednoduchou geometrií pomocí jazyka Structured Query Language (SQL) přes rozhraní SQL Call-Level Interface (SQL/CLI). Dále popisuje práci s prostorovými i neprostorovými atributy prvků.

Prostorové atributy mají hodnotu typu geometrie. Jednoduché prvky (bod, linie, plocha) jsou dvou a méně rozměrné geometrie založené na lineární interpolaci mezi lomovými body. Podle standardu SQL jsou prvky jednoho geometrického typu uloženy jako tabulka prvků, která obsahuje prostorové a neprostorové atributy. Každý prvek je v této tabulce reprezentován jako řádek. Atributy prvků jsou zaznamenány ve sloupcích.

Ukázka formátu WKT pro ukládání dat do PostGIS databáze:

```
(LINESTRING(14.472216796876 50.100783885494, 17.218798828125  
49.604947271051, 18.273486328125 49.853501484927), 102067) (1)
```

První slovo určuje druh prvku (point, linestring, polygon, multibod, multipoint, ...) a v závorce jsou vypsány souřadnice vrcholů. Za závorkou je ještě volitelný parametr kód souřadnicového systému.

Prostorové atributy jsou geometrického datového typu definovaného v tomto standardu pro SQL. Neprostorové atributy mohou být datového typu definovaného tímto

standardem pro SQL anebo definovaného pomocí UDT – uživatelsky definovaný datový typ (Herring, 2010).

Pro tabulky prvků jsou definovány dvě schémata pro zápis geometrie. První schéma se využívá pouze pro SQL předdefinované datové typy. Sloupec popisující hodnotu geometrie je v tabulce geometrií označen jako geometry ID. Tabulka geometrií může být tvořena použitím SQL číselných datových typů nebo SQL binárních datových typů. Druhé schéma pro zápis geometrie využívá podporované geometrické typy (Geometry Types) (Herring, 2010).

2.4 Postup zpracování

Nejdříve bylo potřeba seznámit se s danou problematikou, získat veškeré informace o dostupných webových technologiích a současně tvořit rešerši. Nelehkým úkolem bylo nalézt vhodné technologie a vybrat ty, které budou nejvhodnější pro tento druh aplikace. Důležitým parametrem při výběru byla možnost kombinace jednotlivých metod, protože webová aplikace je často tvořena velkým počtem technologií, kdy každá technologie zastává konkrétní úkolu v daném problému. Bylo dané, že pro mapový server musí být použit UMN MapSever a databáze PostgreSQL s nastavbou PostGIS.

Původně byla myšlenka rozšířit framework p.mapper, který se nakonec ukázal být příliš složitý a propracovaný. Proto byla nakonec zvolena knihovna OpenLayers, která je určená na tvorbu mapových webových aplikací a poskytuje velké množství předdefinovaných funkcí a metod.

Modul začal tvorbou testovacích tabulek v databázovém systému PostgreSQL, které byly naplněny náhodnými daty. Požadavkem byl český souřadnicový systém S-JTSK Křovák, který bohužel většina technologií zatím nepodporuje a proto bylo obtížné nalézt správnou kombinaci parametrů tak, aby data byla v tomto souřadnicovém systému uložena, zároveň správně vizualizována a poté i při dalším přidávání do databáze v tomto souřadnicovém systému ukládána. Databáze byla tvořena postupně v několika krocích, protože bylo potřeba nalézt ideální kombinaci parametrů pro správné fungování modulu. Správný formát databáze je popsán v kapitole 4.4.1 Vytvoření databáze.

Vrstvy se poté vizualizovaly pomocí mapového serveru prostřednictvím webové mapové služby (typ dotazu GetMap, který umožňuje vykreslení mapy) a jeho konfiguračního souboru Mapfile. Zde se určilo, které vrstvy budou viditelné, jakou budou mít symbologii, souřadnicový systém a další parametry. Taková mapa lze zobrazit v prohlížeči jako statický obrázek. Knihovna OpenLayers umožňuje obrázek načíst a poté dynamicky ovládat. Tím je myšlena například změna měřítka, posouvání výřezu, změna viditelných vrstev, aj.

Vedle mapového pole byl vytvořen formulář pro vkládání nových prvků do databáze. Také byla nalezena metoda, která nakreslí do prázdné vektorové vrstvy body, linie či polygony a další metoda, která umí z těchto bodů vzít a vypsát jejich souřadnice ve formátu, který si zvolíme. Nabízí se zde mnoho formátů jako KML, GML, GPX, GeoJSON, nebo WKT (Well-Known Text), což je přímo formát pro ukládání dat do PostGIS databáze.

Po nakreslení nového bodu a vyplnění formuláře, může dojít k přidání prvku do databáze. Formulář metodou POST předává hodnoty polí PHP skriptu, který hodnoty doplní do SQL dotazu a pošle jej databázi. V případě, že nedojde k chybě, prvek je do databáze přidán.

Vypsání atributů jednotlivých prvků je, stejně jako vykreslení mapy, zajištěno mapovým server a webovou mapovou službou, tentokrát je však jako parametr zvolen dotaz GetFeatureInfo. Metodu stejného názvu má také knihovna OpenLayers, která jako jeden z parametrů dostává url mapy. První část volá CGI skript mapového serveru, druhá část je absolutní cestou k Mapfile.

Ukázka URL mapy:

```
http://localhost/cgi-bin/mapserv?  
map=/var/www/Solanska_12/mapfile.map&SRS=EPSG:102067&mode=map
```

Pro editaci atributů jednotlivých prvků byla zvolena šablona mapového serveru, která umožňuje vytvořit jednoduchý formulář z doplněnými hodnotami z databáze. Formulář je propojený s PHP skriptem, který při změně některé z hodnot, SQL dotazem pro aktualizaci (UPDATE) změní záznamy v databázi, popř. smaže prvek z databáze (DELETE FROM).

V konečné fázi bylo navrženo uživatelské rozhraní, vzhled byl ponechán zcela jednoduchý, s možností úpravy vzhledu dle zadání uživatele. Byly mu přidány potřebné funkce, které zamezují vzniku chyb při vkládání do databáze a zajišťují intuitivní ovládání modulu.

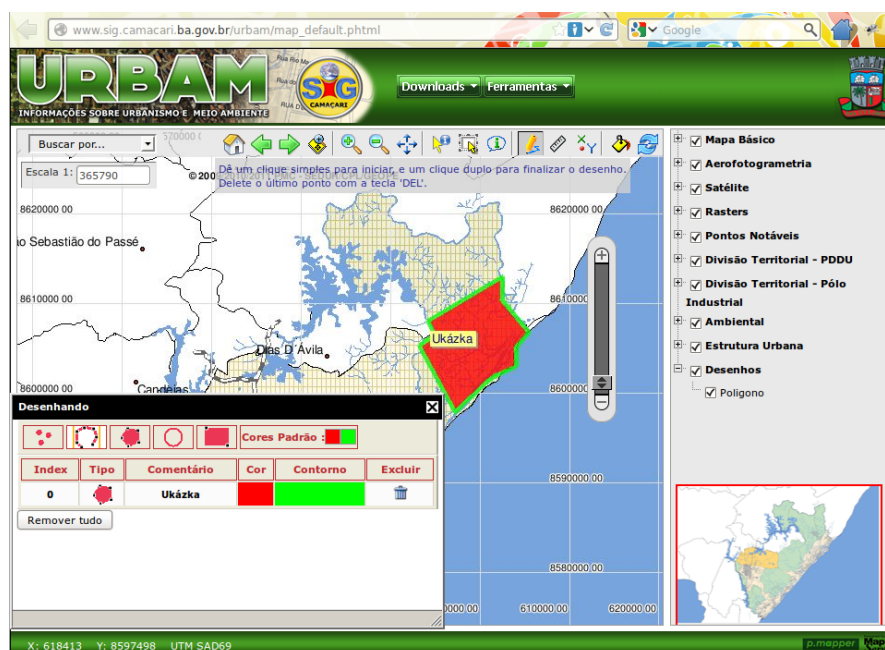
3 SOUČASNÝ STAV ŘEŠENÉ PROBLEMATIKY

Mapové servery se pro prezentaci geografických dat na internetu používají stále častěji. Je možné vytvořit úplně jednoduchou statickou mapu s použitím mapového serveru a HTML kódu, nebo vytvořit dynamickou mapovou aplikaci v kombinaci s programovacími jazyky jako například JavaScript, Ajax, či PHP. Je mnoho možností, jak vytvořit takovou aplikaci. Tím jednodušším způsobem je využití některého z framework dostupných na internetu. Je tím myšlena aplikace připravena pro uživatele se všemi podpůrnými programy a knihovny, doplněná podrobným manuálem pro možnost vlastního nastavení. Framework jsou často rozšiřitelné, takže s potřebnými zkušenostmi je možné vytvořit aplikaci na míru. Také se nabízí možnost naprogramovat své vlastní grafické interaktivní rozhraní.

3.1 Nekomerční řešení

3.1.1 p.mapper

Jedním z již předpřipravených framework je p.mapper, graficky celkem propracovaný framework, který spolupracuje s UMN MapServer a samotné rozhraní je naprogramované v PHP a JavaScript. Použití samotného p.mapper se základními funkcemi není tak ojedinělé, méně časté jsou aplikace, které používají nějaké rozšíření, ať už možnost zadávat dotazy, ukládat body zájmu nebo právě přidávat nové prvky či editovat. Pár příkladů se nachází přímo na oficiálních webových stránkách p.mapper (<http://pmapper.net/gallery.shtml>). Vhodným příkladem, ve kterém je využito rozšíření pro nakreslení nového prvku, je informační server o urbanismu a životním prostředí (viz obr. 1). Není zde však řešeno ukládání ani načítání bodů, ani nebyl nalezen žádný návod, jakým způsobem by ukládání mělo být řešeno.

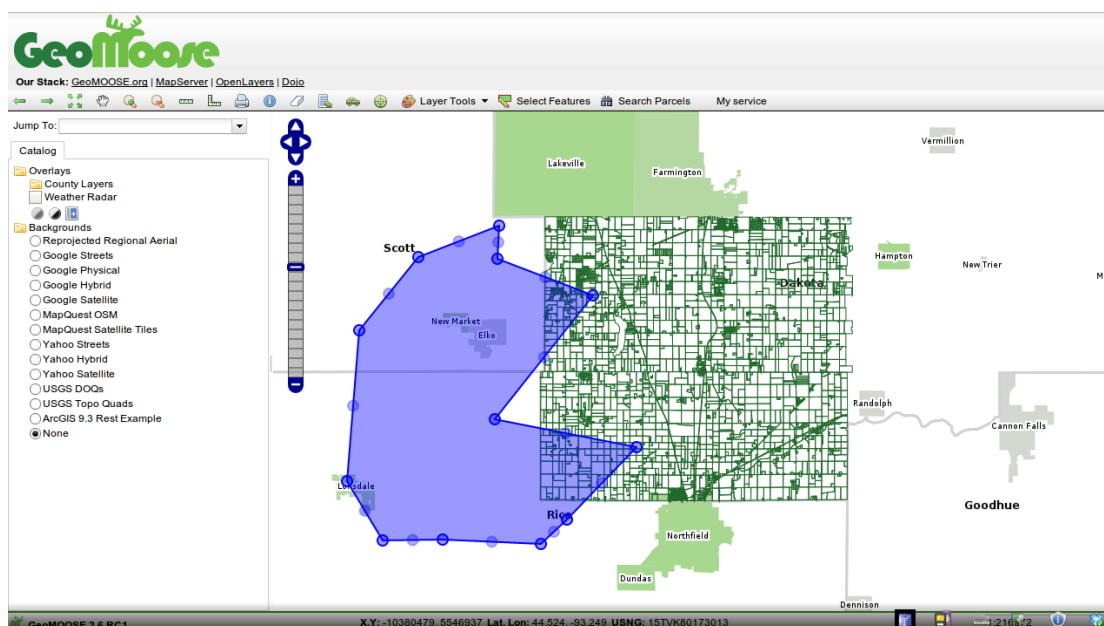


Obr. 1: Kreslení nových bodů pomocí aplikace p.mapper

(zdroj: http://www.sig.camacari.ba.gov.br/urbam/map_default.phtml)

3.1.2 GeoMoose

Další možností, jak snadno vytvořit mapovou aplikaci, je využít framework GeoMoose (viz obr. 2) napsaný v JavaScript, založený na UMN MapServer a knihovně OpenLayers. Na oficiálních stránkách je možné si vyzkoušet, jak aplikace vypadá a funguje. Od nejnovější verze 2.6, která vyšla v dubnu 2012, je k dispozici i návod, kde je popsáno, jakým způsobem aplikaci zprovoznit, včetně založení databáze a všech konfigurací. Vstup a editace je řešena převážně knihovnou OpenLayers. Více informací na www.geomoose.org.



Obr. 2: Kreslení nových bodů pomocí aplikace GeoMoose
(Zdroj: <http://www.geomoose.org>)

3.1.3 Další nekomerční aplikace

Další z příkladů framework, které jsou založeny na UMN MapServer, jsou poměrně mladé projekty ka-Map (viz <http://ka-map.maptools.org/>) a msCross (viz <http://swik.net/msCross>), fungující na straně klienta, naprogramované v knihovně Ajax a JavaScript.

Dále framework založené na principu klient-server:

- MapBender (PHP/ JavaScript, viz <http://en.wikipedia.org/wiki/Mapbender>),
- MapFish (Python/JavaScript, viz <http://mapfish.org/>),
- Cartoweb (PHP/JavaScript, viz <http://www.cartoweb.org/>),
- Fusion-mapping (viz <http://www.jcmb.com/jcmb2/index.php/products/fusion-mapping>),
- MapGuide (viz. <http://mapguide.osgeo.org/>),
- GeoExt (OpenLayers/Ext Js, viz <http://geoext.org/>).

3.2 Komerční řešení

3.2.1 T-MAPY spol. s r.o.,

Když se zaměříme na komerční sféru, nalezneme firmy, které nabízejí již hotové mapové aplikace, které poté přizpůsobí požadavkům klienta. Česká firma T-MAPY spol. s r.o. vyvinula technologii T-MapServer, který je postaven na UMN MapServer, popřípadě ArcIMS. Tento produkt může být využit pro jednoduché zobrazení map až po komplexní práci s grafickými i databázově orientovanými aplikacemi.

Více informací o produktu T-MapServer včetně odkazů na příklady projektů na http://www.tmapy.cz/public/tmapy/cz/_software/twist/_mapy.html.

3.2.2 Help Service – Remote Sensing spol. s.r.o.

Další českou firmou nabízející kromě jiného i tvorbu a správu komplexních mapových aplikací je Help Service – Remote Sensing spol. s.r.o. Jejich produkty jsou založené na open source technologiích, pro zobrazení rastrových i vektorových dat používají UMN MapServer nebo GeoServer, pro uložení dat PostgreSQL/PostGIS, samotná aplikace je naprogramovaná v knihovně OpenLayers nebo ExtJS. Více informací na oficiálním webu <http://www.bnhelp.cz>.

3.2.3 MapSherpa

Opomenout bychom neměli ani na celosvětovou nabídku. Kanadská firma DM Solution Group Inc. nabízí webovou aplikaci MapSherpa, která umožňuje vytvořit si mapu území Kanady a USA. Aplikace je využitelná především pro outdoorové nadšence. Samotnou aplikaci lze vyzkoušet na <http://www.mapsherpa.com/> a více informací o firmě na oficiálních webových stránkách <http://www.dmsolutions.ca/>.

3.3 Diplomové práce

Na katedře Geoinformatiky Univerzity Palackého v Olomouci vzniklo několik diplomových prací zabývajících se UMN MapServer. Diplomová práce, z které bylo především čerpáno při studiu pramenů, byla magisterské práce Evy Alkové, která zpracovala Archeologický informační systém Olomouce. Její prací bylo upravit p.mapper, který je postaven na UMN MapServer, podle potřeb Archeologického centra v Olomouci a dále připravit všechna potřebná data do podoby databáze, aby se daly data prezentovat do webového rozhraní. Pro tuto práci byla zvolena databáze PostgreSQL s rozšířením PostGIS, která umožňuje ukládání geografických dat. Její práce spočívala především v návrhu a vytvoření databáze, tvorby mapového serveru a úpravu p.mapperu podle potřeb zadavatele, včetně úpravy grafické stránky a zajištění uživatelské funkcionality (Alková, 2010).

Dále bych zmínila práci Petry Pietrasové, která vytvořila aplikaci, která umí zapínat a vypínat vrstvy, používat vrstvy přes WMS službu. Umí se v mapě pohybovat, dotazovat se a následné kódy interpretovat a převádět. Dále obsahuje základní funkce jako tisk aktuální stránky, zobrazení nápovědy a legendy (Pietrasová, 2008). Práce byla postavena

na nekomerčním řešení, použity byly UMN MapServer, PHP Mapscript, PHP a JavaScript.

Mapový portál olomouckých parků v roce 2008 zpracoval Petr Šimáček jako svou diplomovou práci. Mapový portál postavil na framework p.mapper, který je založený na UMN MapServer. To zahrnovalo instalaci a konfiguraci všech potřebných podpůrných SW, návrh grafického rozhraní a vlastní přizpůsobení mapového serveru, jeho zasazení do webových stránek (vytvoření mapového portálu o parcích) a nakonec jeho úspěšné spuštění v prostředí Internetu (Šimáček, 2008). Mapová aplikace je k nahlédnutí na webu: <http://gislib.upol.cz/app/simacek08/pmapper/index.htm>.

Cílem diplomové práce Věry Janíkové bylo vytvoření funkčního a prakticky využitelného informačního systému rezervace Ščůrnica pomocí navržených (doporučených) programů. Byl vytvořen mapový server založený na UMN MapServer a implementována desktopová aplikace IS Ščůrnica u správce rezervace Zo ČSOP Kosenka ve Valašských Kloboukách (Janíková, 2007).

Dále byla čerpáno z bakalářské práce Libora Kimpla Prostorové nastavby nekomerčních databází – vstup a správa geoobjektů, který podrobně zpracoval problematiku databází komerčních i nekomerčních řešení. Konkrétně jsem v této práci nastudovala nekomerční databázový systém PostgreSQL, který je zde zpracován jak teoreticky tak s praktickými příklady. Text lze nalézt také na internetové stránce: <http://www.geoinformatics.upol.cz/dprace/bakalarske/kimpl10//vstup.html>.

Mapovým serverem se zabýval ve své bakalářské práci také Martin Minář s Masarykovy Univerzity v Brně. Hlavním cílem jeho práce bylo vytvořit mapovou aplikaci pro zobrazení epidemiologických dat na webu. K tomu měl posloužit vhodný GIS software. Ukázalo se, že je jich dostupné velké množství jak z řad komerčních, tak nekomerčních produktů. Navíc tyto produkty mezi sebou mohou komunikovat a vyměňovat si data pomocí standardizovaných služeb. Mapy všeobecně jsou snadným orientačním prvkem, proto je jich použito i v této práci. Konkrétně byl použit UMN MapServer, ten zpracovává požadavky pomocí „Open Source“ technologií a také skrze ně vykresluje mapu s příslušnými hodnotami (Minář, 2008). Práce přestože rozebírá všechny dostupné varianty je nakonec postavena na nekomerčním řešení, konkrétně používá mapový server UMN MapServer v kombinaci s Apache HTML server, PHP a JavaScript.

4 VÝSLEDKY

Hlavním výsledkem této bakalářské práce je funkční webový modul pro identifikaci dat, vkládání nových prvků a editaci. Modul obsahuje mapové pole a formulář. Mapu lze interaktivně ovládat, tedy měnit měřítko, posouvat výřezy. Formulář obsahuje funkci pro navigaci, pro identifikaci či editaci bodů a formulář pro vkládání nových prvků. Identifikace po kliknutí na některý z prvků zobrazí okno s hodnotami záznamu z databáze. Ve stejném okně je dále možno změnit některý z atributů či celý prvek smazat. Nový záznam do databáze lze vložit po nakreslení prvku a vyplněním polí ve formuláři pro vkládání nových prvků.

Tato kapitola dále popisuje způsob tvorby modulu, její požadavky, konfigurace a použití. Modul vznikl postupně a je kombinací několika skriptovacích jazyků, podle nichž jsou soubory logicky rozděleny do složek. To zajišťuje větší přehlednost a možnost snadno doplnit kód podle potřeb uživatele. Modul je připraven tak, aby se dal podle potřeb rozšířit či upravit.

Uživatelské rozhraní bylo vytvořeno tak, aby obsahovalo všechny potřebné funkce, které může uživatel při editování či vkládání nových prvků potřebovat, a snaží se, aby ovládání modulu bylo co nejvíce intuitivní.

4.1 Programové komponenty

Nejdříve je nutné nainstalovat všechny potřebné programy, moduly a knihovny, přidat do příslušných souborů český souřadnicový systém a pro správné zobrazování upravit absolutní cesty s ohledem na adresářovou strukturu počítače.

Konkrétně je potřeba mít nainstalovány následující programy, na pořadí instalace nezáleží: webový server *Apache HTTP Server*, mapový server *UMN MapServer*, databázi *PostgreSQL* s nastavbou *PostGIS*, dále jazyk *PHP*, kde je potřeba zajistit podporu pro *PostgreSQL*.

V operačním systému Microsoft Windows se jako nejsnazší cesta nabízí stažení balíku *ms4w* (aktuální verze 3.0.6), který obsahuje *Apache HTTP Server*, *PHP*, *UMN MapServer* a další programy. Stažení balíku a seznam všech programů, které obsahuje, je možné získat na oficiálních stránkách <http://www.maptools.org/ms4w/>. Dále se musí nainstalovat *PostgreSQL* (aktuální verze 9.1.4), které při instalaci nabízí i možnost nainstalování nastavby *PostGIS* (verzi 1.5.3).

4.2 Struktura modulu

Modul se tedy skládá z několika skriptů, které jsou logicky členěny do složek (viz obr. 3), převážně podle použitého skriptovacího jazyku. Tuto složku je potřeba přepokopírovat do složky stanovené nastavením volby `DocumentRoot` uvedené v konfiguračním souboru `httpd.conf` webového serveru Apache. V kořenovém adresáři se nachází dva skripty *index.php* a *mapfile.map*.

Soubor *index.php* spouští celou aplikaci, obsahuje metadata html stránky, odkazuje na soubory definující styl html stránky, soubor *OL_Draw.js* a na všechny knihovny (OpenLayers, JS Validate). Dále je zde definován formulář pro výběr funkcionality a formulář pro vložení nového bodu, který dále odkazuje na PHP skript.

Soubor *mapfile.map* se stará o vzhled prvků v mapě, určení souřadnicových systémů, viditelných vrstev, symbologie, popisu, nastavení pro přenos přes wms službu a další.

📁 Solanska_12	5 položek	Složka
📁 js	3 položky	Složka
📁 jsvalidate	3 položky	Složka
📁 OpenLayers-2.11	8 položek	Složka
📄 OL_Draw.js	9,1 kB	Program v JavaScriptu
📁 php	5 položek	Složka
📄 sql_connect.php	311 bajtů	Skript PHP
📄 sql_delete.php	1,4 kB	Skript PHP
📄 sql_inputFromForm.php	1,0 kB	Skript PHP
📄 sql_insert.php	1,1 kB	Skript PHP
📄 sql_update.php	1,5 kB	Skript PHP
📁 style	8 položek	Složka
🖼️ hlavicka_apl.png	19,9 kB	Obrázek PNG
📄 mapstyle.css	1,6 kB	Styl CSS
🖼️ paticka_apl.png	9,4 kB	Obrázek PNG
🏠 polygon.ico	5,4 kB	Ikona Microsoft
📄 question.jpeg	444 bajtů	JPEG Image
📄 template_line.html	1,3 kB	Dokument HTML
📄 template_point.html	1,7 kB	Dokument HTML
📄 template_polygon.html	1,8 kB	Dokument HTML
📄 index.php	10,1 kB	Skript PHP
📄 mapfile.map	5,0 kB	Prostý textový dokument

Obr. 3: Adresářová struktura modulu

Složka s názvem *js* obsahuje všechny skripty jazyka JavaScript. Aby modul fungoval v offline režimu, je v této složce uložena knihovna OpenLayers a knihovna pro validaci JS Validate, díky které je validace formuláře velice snadno nastavitelná. Validací je myšleno především zajištění, že nebude provedeno odeslání hodnot do databáze dříve, než budou vyplněna všechna pole. Důležitým souborem je *OL_Draw.js*, který zajišťuje především vykreslení samotné mapy, její ovládání a možnost přidání a editace prvků. Soubor *OL_Draw.js* je více rozebrán v kapitole 4.3 Popis fungování modulu.

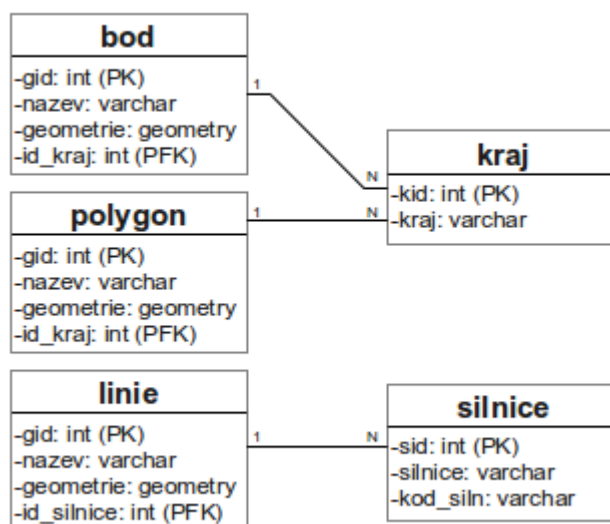
Všech pět PHP skriptů je uloženo ve složce *php*, která obsahuje samostatný soubor *php_connect.php* s nastavením připojení do databáze a soubor *php_inputFromForm.php*, který metodou POST přebírá hodnoty z polí a ukládá je do proměnných jazyka PHP. Tyto dva soubory jsou dále volány všemi dalšími skripty. Všechny tyto skripty obsahují

jeden z SQL příkazů, které zajišťují přidání nového prvku do databáze, aktualizaci prvku a smazání prvku. (*php_insert.php, php_update.php, php_delete.php*)

Další složka, nazvaná *style*, obsahuje všechny skripty související se vzhledem modulu. Tedy hlavně soubor *mapstyle.css* a šablony mapového serveru, které nastavují vzhled vyskakovacích oken.

4.3 Struktura databáze

Byla navržena databázová struktura o pěti tabulkách a naplněna fiktivními daty. Názvy tří z tabulek odpovídají základním prvkům bod, linie a polygon, které mohou být zobrazovány v mapě. Další dvě tabulky představují atributy, jako příklad byly zvoleny tabulky silnice pro linie a kraj pro bod a polygon. Na těchto tabulkách bylo testováno převážně propojení tabulek a dotazování (viz obr. 4).



Obr. 4: Struktura databáze

4.4 Tvorba modulu

4.4.1 Vytvoření databáze

Vytváření geoprostorové databáze má svá pravidla a specifika. Každá vytvořená databáze musí obsahovat tabulku souřadnicových systémů a tabulku geometrie (*geometry_column* a *geography_column*)¹. Tabulky se do databáze přidají následujícím postupem.

Konzolový příkaz pro přidání tabulek:

```
psql -d název_databáze -f cesta_k_souboru (2)
```

Umístění souborů v operačním systému Linux:

```
/usr/share/postgresql/9.1/contrib/postgis-1.5/postgis.sql
/usr/share/postgresql/9.1/contrib/postgis-1.5/spatial_ref_sys.sql
```

1 Tabulky jsou ve formátu sql.

Umístění souborů v operačním systému Microsoft Windows:

```
/Program Files/Postgresql/9.1/contrib/postgis-1.5/postgis.sql  
/Program Files/Postgresql/9.1/contrib/postgis-1.5/spatial_ref_sys.sql
```

Český souřadnicový systém je potřeba přidat do tabulky souřadnicových systémů *spatial_ref_sys*. Do databáze PostGIS jej přidáme pomocí příkazu napsaného níže, ať už prostřednictvím programu pgAdmin III nebo přímo přes příkazový řádek databáze.

Příkaz pro přidání českého souřadnicového systému do PostGIS:

```
INSERT INTO "spatial_ref_sys"  
("srid","auth_name","auth_srid","srtext","proj4text")  
VALUES (102067, 'ESRI', 102067, 'PROJCS["S-  
JTSK_Krovak_East_North",GEOGCS["GCS_S_JTSK",DATUM["D_S_JTSK",SPHEROID  
["Bessel_1841",6377397.155,299.1528128]],PRIMEM["Greenwich",0],UNIT["  
Degree",0.017453292519943295]],PROJECTION["Krovak"],PARAMETER["False_  
Easting",0],PARAMETER["False_Northing",0],PARAMETER["Pseudo_Standard_  
Parallel_1",78.5],PARAMETER["Scale_Factor",0.9999],PARAMETER["Azimuth  
",30.28813975277778],PARAMETER["Longitude_Of_Center",24.8333333333333  
3],PARAMETER["Latitude_Of_Center",49.5],PARAMETER["X_Scale",-  
1],PARAMETER["Y_Scale",1],PARAMETER["XY_Plane_Rotation",90],UNIT["Met  
er",1]]','+proj=krovak +lat_0=49.5 +lon_0=24.83333333333333  
+alpha=30.28813975277778 +k=0.9999 +x_0=0 +y_0=0 +ellps=bessel  
+units=m');
```

 (3)

Při vytváření každé tabulky je potřeba nadefinovat sloupec s jednoznačným identifikátorem, názvem prvku a geometrie, další atributy jsou volitelné.

Ukázka vytvoření nové tabulky:

```
CREATE TABLE bod (  
gid serial NOT NULL PRIMARY KEY,  
nazev character varying(30) NOT NULL UNIQUE,  
geometrie geometry  
) WITH (OIDS=FALSE);
```

 (4)

Vkládání jednotlivých geoprvků se provádí příkazem INSERT a musí být dodržen následující tvar:

```
INSERT INTO bod (nazev, geometrie) VALUES ('bod1',  
St_GeomFromText('POINT(72 27)',102067));
```

 (5)

Při vkládání nového prvku se jednoznačný identifikátor (gid) přidává automaticky vždy o 1 vyšší než nejvyšší dosud použitý (to zajišťuje datový typ serial). Vyplnit se musí atributy název a geometrie. Geometrie se vkládá ve formátu WKT, který určí, zda přidávaný prvek bude bod, linie nebo polygon, a dále v závorce souřadnice vrcholových bodů a souřadnicový systém.

4.4.2 PHP skript

Připojení k databázi je zajištěno příkazem *pg_connect*, podobně jako odpojení od databáze *pg_close*. V parametrech těchto příkazů se zadává identifikace serveru, název databáze, název vlastníka databáze a heslo.

Po odeslání html formuláře se zavolá PHP skript, který nejdříve zjistí, zda již existuje takový prvek v databázi. Aby byl SQL příkaz proveden, je potřeba zavolat funkci *pg_query*, která se prvním parametrem připojí k databázi a druhým parametrem je samotný SQL příkaz.

Způsob připojení k databázi a zjištění existence prvku v databázi:

```
$dbh = pg_connect ("host=localhost dbname=geodatabase
  user=postgres password=***");
$select = pg_query($dbh, "SELECT * FROM bod where nazev='Praha'");
$result_empty = (pg_num_rows($select) == 0);
//pokud je výsledek 0, daný prvek v databázi existuje (6)
```

Na základě výsledku toho dotazu se poté prvek buď přidá do databáze nebo se upraví některý z jeho atributů (viz ukázka kódu č. 7). Podobně je zajištěno také smazání prvku z databáze jen použit příkaz DELETE FROM.

```
if (!$result_empty) {
  $sql = "UPDATE bod SET nazev = „Brno“ where gid=1";
} else {
  $sql = "INSERT INTO bod (nazev, geometrie, kraj) VALUES
    ('Brno', ST_GeomFromText('POINT(-598523 -1159726),102067),
    'Jihomoravský');";
} (7)
```

4.4.3 Struktura MapServer template

V této aplikaci byly šablony mapového serveru využity pro formulář, který slouží pro identifikaci prvků, a dále pro úpravu hodnot prvků, tedy pro jejich aktualizaci v databázi (viz obr. 8: Okno s identifikací prvku a možností editace).

Šablona lze vytvořit pro celý Mapfile, nebo pro jednotlivé vrstvy. V tomto případě byly vytvořeny tři šablony pro každý z prvků.

Aby bylo zřejmé, že se jedná o šablonu, je potřeba vždy na první řádek napsat:

```
<!-- MapServer Template -->
```

Pro vypsání hodnoty stačí název příslušného atributu napsat do hranatých závorek:

```
gid:[gid] nazev: [nazev]
```

Příklad možného výsledku:

```
gid: 1 nazev: Praha
```

Aby bylo možné editovat atributy, byl do souboru přidán jednoduchý formulář, který při kliknutí na prvek, doplní hodnoty z databáze do formuláře a nechá pole přístupná pro úpravu.

```
<form name="theform" action="sql_update.php" method="post">
  Bod GID: <input name="gid" type="text" value="[gid]" /><br />
  Nazev: <input name="nazev" type="text" value="[nazev]" /><br />
  Kraj: <select name="kraj">
    <option value="[id_kraj]">[kraj]</option>
    <option value="1">Hlavní mesto Praha</option>
    <option value="3">Jihoceský</option>
    <option value="11">Jihomoravsky</option>
    <option value="5">Karlovarsky</option>
  </select><br />
  <input name="submit" type="submit" value="Upravit prvek" />
</form>
```

Po kliknutí na tlačítko „Upravit prvek“ se zavolá PHP skript a data se na základě formuláře upraví SQL příkazem UPDATE. Stejně funguje i tlačítko „Smazat prvek“ jen se zavolá SQL příkaz DELETE FROM.

4.4.4 Skript OpenLayers

Propojení html kódu *index.php* se skriptem *OL_Draw.js* probíhá prostřednictvím tagu *div*, který je definován v html kódu. Jeho id je poté voláno skriptem *OL_Draw.js* objektem *OpenLayers.Map*. Pro snazší pochopení je přidána ukázka obou kódů.

[index.php](#):

```
<div id="map"></div>
```

[OL_Draw.js](#):

```
var options = {
  controls:[
    new OpenLayers.Control.PanZoom(), // zoom
    new OpenLayers.Control.MousePosition(), // pozice myši
    new OpenLayers.Control.OverviewMap(), // přehledová mapka
  ],
  maxExtent: bounds,
  projection: "epsg:102067", //definice systému S-JTSK Křovák
  ...
};
var map = new OpenLayers.Map("map", options); // prvním parametrem je
id tagu div, druhým nastavení vzhledu
```

Hlavním objektem knihovny OpenLayers je *OpenLayers.Map*, kterému jsou předávány dva parametry. Prvním je *div* id a druhým nastavení, konkrétně zvolení souřadnicového systému, hraničních bodů pro výřez mapy, ovládacích prvků jako je zoom, posouvání mapy, zobrazování aktuální souřadnice v závislosti na pohybu myši.

Dalším krokem je nadefinování vrstev. Jak již bylo dříve zmíněno, mapa, která se načítá, je generována mapovým serverem a předávána webovou mapovou službou. Obrázek, resp. mapu načtenou knihovnou OpenLayers, je možno přibližovat či posouvat výřezy.

Pro možnost kreslení nových prvků, byla zadefinována také prázdná vektorová vrstva, která se používá pro přechodné uložení souřadnic. Vždy, když se vkládá nový prvek nebo prvek jiného typu, se vrstva vyčistí. Konkrétně to nastane například při změně vkládání z linie na polygon. Pak dojde k vymazání všech prvků vektorové vrstvy.

Pro kreslení nových prvků byla zvolena metoda *DrawFeature()*, která jako parametr dostává právě prázdnou vektorovou vrstvu, kde může dočasně ukládat, a poté třídu pro kreslení prvku (*OpenLayers.Handler.Point*). V případě, že uživatel vytvoří linii, která představuje dálnici z Prahy do Brna, a nepodaří se mu všechny vrcholy umístit na správná místa, pak se zde nabízí možnost prvek ještě před přidáním do databáze upravit. Umožňuje to metoda *ModifyFeature.RESHAPE*. Ta po kliknutí na již nakreslený prvek zobrazí všechny vrcholy a umožní s nimi manipulovat, popř. vytvořit další vrcholy.

Postup získání souřadnic z vektorové vrstvy a jejich zapsání do pole Geometrie ve formuláři je prováděna v několika krocích. Nejdříve je proveden výběr nakresleného prvku metodou *SelectFeature()*, která jako parametr dostává především naplněnou vektorovou vrstvu.

Zapsání souřadnic nově vzniklých prvků do formuláře vyžadovalo naprogramování vlastních funkcí *updateFormats()* a *serialize()*. Funkce *updateFormats()* určuje, v jakém formátu a v jakém souřadnicovém systému se souřadnice zapíše do pole Geometrie. Funkce *serialize()* zajišťuje, že souřadnice se zapíše do formuláře, který je určen názvem.

Funkce *updateFormats()* a *serialize()*:

```
function updateFormats() {
    var in_options = { //určení souradnicoveho systemu
        'internalProjection': map.baseLayer.projection,
        'externalProjection': new OpenLayers.Projection("EPSG:102067")
    };

    var out_options = {
        'internalProjection': map.baseLayer.projection,
        'externalProjection': new OpenLayers.Projection("EPSG:102067")
    };

    formats = { // určení formátu WKT
        'in': { wkt: new OpenLayers.Format.WKT(in_options), },
        'out': { wkt: new OpenLayers.Format.WKT(out_options), }
    };
}

function serialize(feature) {
    var type = "wkt";
    var pretty = 1;
    var str = formats['out'][type].write(feature, pretty);
}
```

```

    str = str.replace(/,/g, ', ');
    document.forms['theform'].geometrie.value = str;
}

```

Ukázka definování hlavičky formuláře a pole Geometrie (soubor index.php):

```

<form name="theform" action="<? echo $_SERVER['PHP_SELF'];?>"
method="POST">
<input type="text" name="geometrie" id="geomText" readonly="readonly"
size="37" class="jsrequired" disabled />

```

Další důležitou funkcionalitou je identifikace prvků v mapě a úprava hodnot jejich atributů. Identifikace bylo dosaženo metodou *WMSGetFeatureInfo()*, která, jak už z názvu vyplývá, probíhá přes webovou mapovou službu a jako parametr je předávám dotaz *GetFeatureInfo*. V praxi to znamená hlavně snadné dotazování na názvy atributů, které lze využít při tvorbě šablony, která byla podrobně popsána v kapitole 4.2.3. Struktura MapServer template. Metoda *WMSGetFeatureInfo()* dostává několik parametrů. Rozhodující je parametr url mapy a také způsob, jakým dané informace zobrazí. Zvoleno bylo tzv. „popup window“, které je html stránkou, jejíž vzhled a obsah lze nastavit dle požadavků. Hlavním účelem je samozřejmě zobrazení atributů, což bylo blíže popsáno v kapitole 4.2.3. Struktura MapServer template.

Definování metody WMSGetFeatureInfo() pro identifikaci prvků:

```

info: new OpenLayers.Control.WMSGetFeatureInfo({
  url: 'http://localhost/cgi-bin/mapserv?
map=/var/www/Solanska_12/mapfile.map&SRS=EPSG:102067&',
  title: 'Identifikace po kliknutí',
  queryVisible: true,
  hover: false,
  eventListeners: {
    getfeatureinfo: function(event) {
      map.addPopup(new OpenLayers.Popup.FramedCloud(
        "cloud",
        map.getLonLatFromPixel(event.xy),
        null,
        event.text,
        null,
        true
      ))
    }
  }
})

```

Aby nebylo spuštěno několik funkcí zároveň, byla vytvořena funkce *toggleControl()*, která na základě vybraného pole z formuláře, spustí příslušnou funkci. V ukázce výše začíná metoda slovem *info*, což je název, podle kterého funkce *toogleControl()* vybírá, která z funkcí bude aktuálně používána. Ostatní funkce se věnují ovládání modulu ve smyslu vybrání funkcionality, zda se bude posouvat výřez mapy, identifikovat body, či vkládat, když vkládat, jestli bod, linie nebo polygon a když bod, poté at' je možno

zapisovat do polí Název, Geometrie a Kraj, když linie, ať je možno zapisovat do polí Název, Geometrie a Silnice. Funkcionalita je blíže popsána v kapitole 4.6 Popis fungování modulu.

4.5 Konfigurace

Český souřadnicový systém S-JTSK Křovák, který zatím ještě používá ESRI kód 102067, je potřeba vložit do souboru souřadnicových systémů mapového serveru, které jsou uloženy v souboru *epgs*², kde je nutné přidat následující řádek definující parametry souřadnicového systému:

```
#S-JTSK / Krovak 102067
<102067> +proj=krovak +lat_0=49.5 +lon_0=24.83333333333333
+alpha=30.28813972222222 +k=0.9999 +x_0=0 +y_0=0 +ellps=bessel
+pm=greenwich +units=m +no_defs<>
```

4.5.1 Nastavení absolutních cest

Soubor *mapfile.map* si před spuštěním žádá změnu absolutních cest, konkrétně v sekci WEB a v sekci METADATA každé vrstvy.

První řádek, který se musí upravit v sekci WEB, začíná příkazem *IMAGEPATH*. Tím se definuje cesta k dočasnému uložení obrázků, v tomto případě mapy, která je přenášena webovou mapovou službou. Podobně se definuje *IMAGEURL*, který určuje URL adresáře s obrázky. Je možno zvolit jakoukoliv složku, kde nejsou nastavená práva a můžou do něj vstupovat všichni uživatelé (Čepický, 2005).

Další řádek se týká METADAT v sekci WEB, kde je potřeba změnit cestu, která je definována příkazem „*wms_onlineresource*“. První část cesty spouští CGI skript *mapserv*, který v systému Microsoft Windows má navíc příponu *.exe*. Soubor také může být uložen v jiné složce, záleží na způsobu instalace. Přesto při zanechání původního nastavení adresa bývá *http://localhost/cgi-bin/mapserv?* pro Linux a *http://localhost/cgi-bin/mapserv.exe?* pro Microsoft Windows. Druhá část je absolutní cestou k souboru *mapfile.map*, cesta tedy závisí na adresářové struktuře počítače.

Ukázka kódu, který je potřeba opravit, než se modul spustí:

```
WEB      # nastavení parametru pro WMS
IMAGEPATH "/var/www/tmp/"
IMAGEURL  "/tmp/"
METADATA
    "wms_title"          "WMS Krovak Mapa"
    "wms_onlineresource" "http://localhost/cgi-bin/mapserv?
map=/var/www/Solanska_12/mapfile.map&"
...
END
END
```

2 Cesta k souboru epsg: v operačním systému Linux */usr/share/proj/epsg*, v operačním systému Microsoft Windows *\$MS4W_DIR\proj\bin\epsg*, kde *\$MS4W_DIR* je instalační složka balíku *ms4w*

Stejně tak je potřeba upravit název souboru *mapserv*, resp. *mapserv.exe*, podle použitého operačního systému, v sekci METADATA každé definované vrstvy:

```
LAYER
  NAME 'linie'
  ...
  METADATA # nastavení parametru pro WMS a dotazování
    "wms_title"          "linie"
    "wms_onlineresource" "http://localhost/cgi-bin/mapserv?"
  ...
  END
END
```

V souboru *OL_Draw.js* je také potřeba úpravy absolutní cesty, stejně jako v souboru *mapfile.map*, a to na dvou místech popsaných níže.

Při definování zobrazované vrstvy:

```
var layer_sjtsk = new OpenLayers.Layer.WMS(
  "MapServer WMS",
  "http://localhost/cgi-bin/mapserv?",
  map=/var/www/Solanska_12/mapfile.map&SRS=EPSG:102067&",
  ...
);
```

Při dotazování na prvky dotazem GetFeatureInfo:

```
info: new OpenLayers.Control.WMSGetFeatureInfo({
  url: 'http://localhost/cgi-bin/mapserv?
  map=/var/www/Solanska_12/mapfile.map&SRS=EPSG:102067&',
  title: 'Identifikace po kliknutí',
  ...
})
```

4.5.2 Nastavení souboru Mapfile

Konfiguračním souborem mapového serveru je soubor Mapfile (přípona *.map*), který definuje vztahy mezi objekty, definuje jaká data se budou zobrazovat, jaká bude jejich symbologie, popis, souřadnicový systém, aj.

Důležité pro každou vrstvu je definování typu geoprisku a spojení. Definují se následující parametry:

- TYPE odpovídá typu geometrie uložené do databáze, (TYPE [annotation|chart|circle|line|point|polygon|raster|query])
- CONNECTIONTYPE určuje typ spojení, v tomto případě je vybrán POSTGIS, tabulky však mohou být načítány i z databáze Oracle nebo SDE, nebo pomocí WMS, či WFS, (CONNECTIONTYPE [local|ogr|oraclespatial|plugin|postgis|sde|union|wfs|wms])

- CONNECTION se definuje různě pro typy spojení, pro PostGIS je v parametrech spojení název domény localhost, číslo portu, název databáze, uživatel a heslo,
- DATA je v podstatě SQL příkaz SELECT v následujícím formátu:

```
SELECT nazev_sloupce      (ve kterém je uložena geometrie)
FROM nazev_1tabulky
JOIN* nazev_2tabulky ON atribut.1tabulka=atribut.2tabulka
USING UNIQUE id          (název jednoznačného id)
USING srid=102067        (souřadnicový systém)
```

*join se volitelný parametr

Příklad spojení pro bod:

```
TYPE POINT
CONNECTIONTYPE POSTGIS
CONNECTION "host=localhost port=5432 dbname=bp user=postgres
password=***"
DATA "geometrie from bod join kraje USING UNIQUE gid USING
srid=102067"
```

Bylo potřeba dát si pozor i na první pohled nedůležité detaily, jak je velikost písma u definování metadat jednotlivých vrstev, která umožňují identifikaci prvků. Například malá písma se píšou u dat, která jsou ve formátu ESRI shapefile, kdežto velká písmena se píšou při načítání z databáze.

Příklad metadat u záznamů z databáze:

```
METADATA
"RESULT_FIELDS" "gid, nazev, kraj" #nazvy atributu v databazi
"RESULT_HEADERS" "GID, Jmeno, Kraj" #zobrazovane nazvy v aplikaci
"LAYER_ENCODING" "UTF-8"
END
```

Parametry pro přenos dat prostřednictvím webové mapové služby je třeba nadefinovat v sekci WEB METADATA. Příkazy začínají v závislosti na zvoleném typu přenosu, tedy například wms, wfs, ows, gml. V tomto případě začínají wms a definuje se název, souřadnicový systém, typy přenosu a příkazem *wms_onlineresource* spuštění CGI skriptu mapového serveru a cesta k souboru Mapfile.

Příklad definování parametrů pro wms:

```
WEB
IMAGEPATH "/var/www/tmp/"
IMAGEURL "/tmp/"
METADATA
"wms_title" #nazev mapy
  „WMS Krovak Mapa"
"wms_onlineresource" #cesta k mapfile
  "http://localhost/cgi-bin/mapserv?map=mapfile.map&"
"wms_srs" #urceni souradnicoveho souboru
```



```

        "EPSG:102067"
        "wms_enable_request"    #typ dotazu
        "GetMap GetFeatureInfo"
    END
END

```

4.6 Popis fungování modulu

Databáze uchovává data ve formě tabulky s definovanými atributy včetně atributu s geometrií prvku. Mapový server zajišťuje vykreslení mapy, data pro výslednou mapu jsou brána z databáze, a její přenos je zprostředkován webovou mapovou službou. Vygenerovaná mapa má své URL, které je pak načítáno knihovnou OpenLayers. URL se skládá ze dvou částí, první je volání vlastního CGI skriptu MapServer a druhá část, začínající příkazem „map=“, je cesta k Mapfile s parametry rozhodující o zpracování mapy, příklad parametrů: vrstva (LAYER), souřadnicový systém (SRC), ohraničení výřezu (BBOX), typ dotazu (REQUEST), způsob zobrazení (MODE), verze (VERSION) a další.

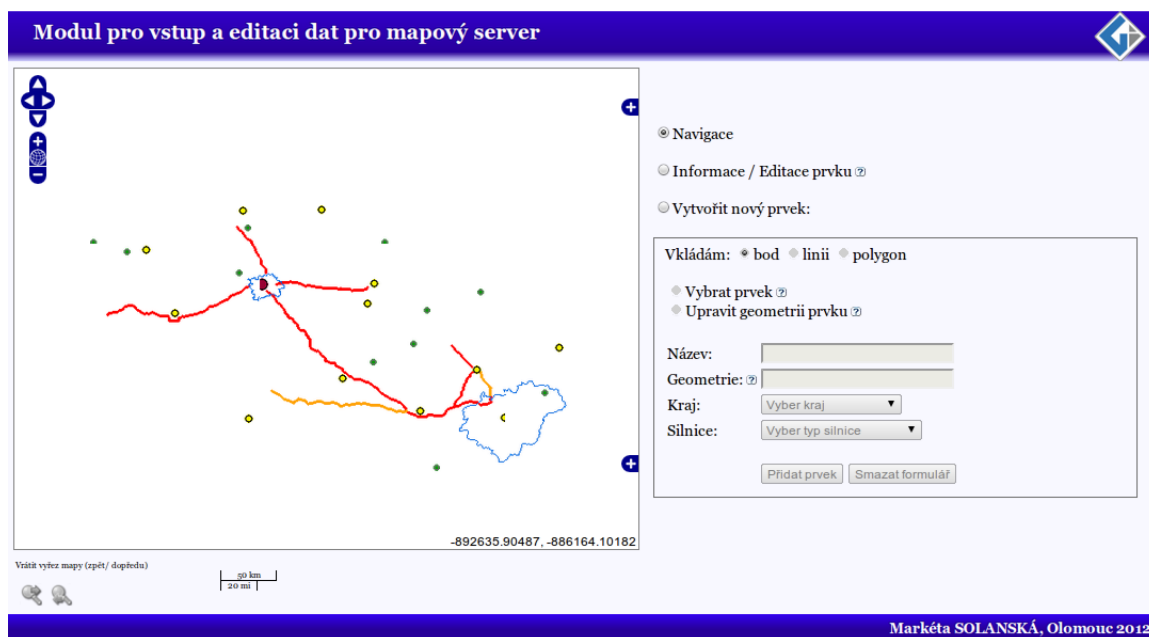
OpenLayers vytvoří mapové pole, vloží je do mapovým serverem a přidá mu funkcionalitu jako je zoom, posouvání výřezu, změna vrstev, navigace, zobrazení měřítka a další. Stejná knihovna dále zajišťuje také nakreslení nového bodu, či editaci stávajících bodů. O to se starají metody *DrawFeature()*, *ModifyFeature()*, *WMSGetFeatureInfo()*.

Dále se pomocí funkce *serialize()* a *updateFormats()* získají souřadnice ve formátu WKT, což je formát pro PostGIS. Souřadnice se zapíší do formuláře do pole Geometrie. Po doplnění dalších atributů je možné odeslat celý html formulář metodou POST PHP skriptu. Ten hodnoty z formuláře uloží do proměnných a ve formě SQL dotazu jej předá databázi.

Po přidání nového či editovaného prvku do databáze se prvek objeví v mapě po obnovení prohlížeče a je možno s ním dále manipulovat.

Bylo navrženo velice jednoduché uživatelské rozhraní (viz obr. 5), aby modul mohl být začleněn do jakéhokoliv webu či portálu. Byly použity pouze základní prvky definující vzhled. Modul tvoří jedna webová stránka, vlevo je mapové pole a vpravo ovládání celého modulu. Byla snaha vytvořit aplikaci tak, aby bylo ovládání intuitivní, přesto jako u každého modulu je třeba si ovládání vyzkoušet a zjistit, jak která funkce funguje.

Mapové pole, umístěno vlevo, obsahuje ovládací prvky pro zoom, změnu viditelných vrstev a přehledovou mapku. Pod mapou je vlevo historie přiblížení, takže šipkami vpřed a vzad lze snadno vrátit výřez z předchozího stavu. Napravo od historie je měřítko.



Obr. 5: Ukázka základního rozložení modulu

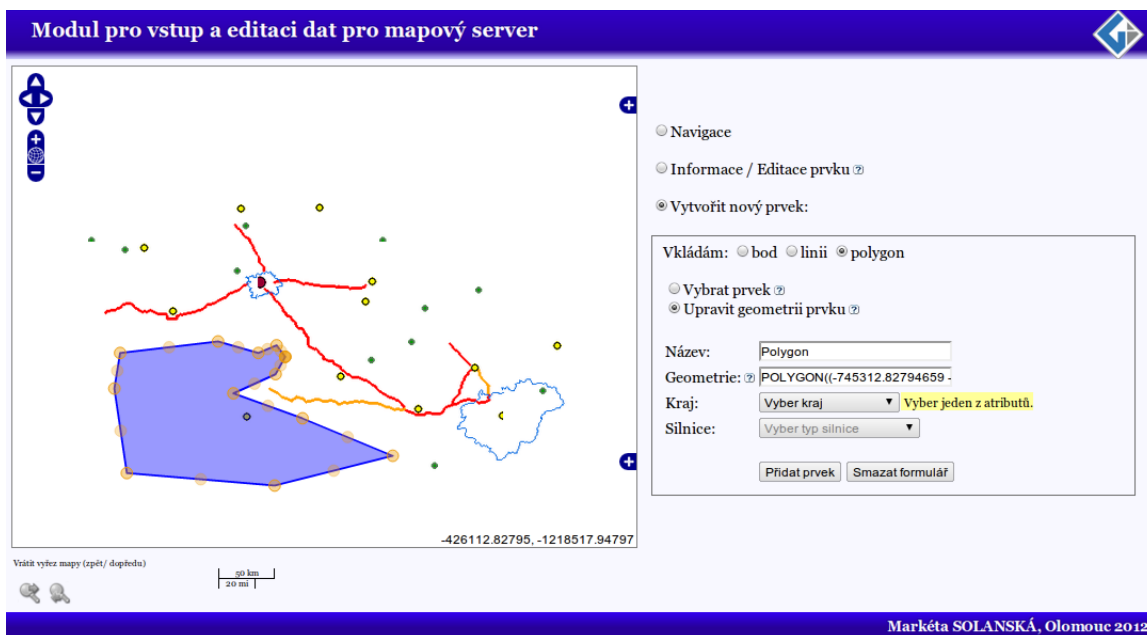
Větší důraz než na vzhled byl tedy kladen na jednoznačné oddělení funkčnosti modulu. Uživatel může zvolit pouze jednu z možností:

- navigace,
- informace / editace prvku,
- vytvořit nový prvek.

„Navigace“ umožňuje posouvání výřezu mapy, „Informace / Editace prvku“ po kliknutí do mapy vykreslí okno, ze kterého je možné získat hodnoty z databáze, nebo upravit hodnoty jeho atributů. Poslední možnost „Vytvořit nový prvek:“ nabízí dále možnost zvolit, zda se bude vkládat bod, linie nebo polygon.

Podle zvoleného prvku se potom stanou aktivní ta pole, která přísluší danému prvku. Například u prvku bod je atributem kraj, kdežto u prvku linie je atributem silnice. Proto při vkládání bodu bude pole kraj možno měnit, kdežto pole silnice zůstane needitovatelné. Formulář je také před odesláním do databáze validován, aby nedošlo k předání nevyplněného pole. Věta „Vyplň toto pole.“ se zobrazí při pokusu odeslat formulář za nevyplnění některého z polí.

Možnost kreslení nového prvku lze již při kliknutí na jeden z prvků bod, linie, polygon. Pokud má uživatel potřebu změnit tvar prvku, ještě před odesláním do databáze, může kliknout na „Upravit geometrii pole“ a po kliknutí na prvek vrcholy přesunout, či vytvořit nové (viz obr. 6).



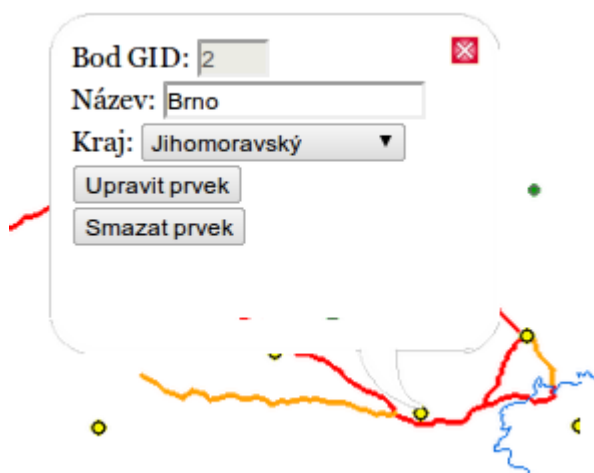
Obr. 6: Ukázka úpravy geometrie ještě před přidáním do databáze

Kreslení umožňuje nakreslit nekonečně mnoho prvků, proto když je prvek hotový, je potřeba kliknout na „Vybrat prvek“. Po vybrání prvku (po najetí myši) se souřadnice prvku, který má být přidán, zapíše do pole Geometrie (viz obr. 7).

Geometrie:

Obr. 7: Ukázka zapsání souřadnic do pole Geometrie ve formátu WKT

Po vyplnění všech polí je stisknuto tlačítko „Přidat“ a nový prvek se přidá do databáze. Po obnovení stránky se vykreslí do mapového pole.



Obr. 8: Okno s identifikací prvku a možností editace

Funkce „Informace / Editace prvku“ umožňují uživateli identifikovat prvky, tedy kliknout na jakýkoliv prvek v mapě. Výsledkem je okno, které zobrazí hodnoty atributů vybraného prvku (viz obr. 8).

Upravovat můžeme jeho atributy přepsáním hodnot v polích a stisknutím tlačítka „Upravit prvek“ je změnit. V případě nelibosti lze jednoduchým „Smazat prvek“, prvek vymazat z databáze.

5 DISKUZE

Tvorba jakékoliv aplikace se neobejde bez problémů, zvlášť pokud jde o propracovanou aplikaci, kde k dosažení výsledku je potřeba použít několik druhů technologií. Naučit se programovat v jakémkoliv programovacím nebo skriptovacím jazyku je časově náročné a ani pak není zajištěno, že krátký kus kódu, který se někdy napíše za pár minut, se nebude psát hodiny.

V první fázi práce bylo potřeba vybrat programové prostředky, které budou použity pro vytvoření tohoto modulu. PostgreSQL bylo jedním z požadavků na tvorbu modulu. Samozřejmě mohla být zvolena i jiná databáze s podporou prostorových dat. MySQL umožňovalo ukládání prostorových dat pomocí nástavby SpatialExtension, ale jeho vývoj nyní stagnuje a je závislý na zájmu firmy Oracle. Novou databází je SQLite s nástavbou SpatialLite a RasterLite, která je stále ve fázi vývoje.

Nástavba PostGIS umožňuje uložit souřadnice do dvou různých datových typů geography a geometry. Každý z nich má svá pozitiva i negativa. Typ geography ukládá souřadnice v kartézských rovinných souřadnicích, kterému odpovídá souřadnicový systém WGS84. V případě, že budeme chtít získat vzdálenost dvou bodu (příkazem ST_Distance), typ geography jako výsledek vypíše nejkratší vzdálenost v rovině v kilometrech. Typ geometry ukládá data v polárním rovinném systému a umožňuje nastavit souřadnicový systém podle potřeb. Výsledkem dotazu na vzdálenost dvou bodů bude úhel ve stupních. Po převodu do metrické soustavy tedy dostaneme nejkratší vzdálenost na kouli (OpenGeo, 2012b).

Přestože se zdá geography výhodnější, typ geometry obsahuje více funkcí, k tomu souřadnice se dají jednoduše převést do rovinných souřadnic pomocí funkce (Geography(ST_Transform)). Typ geography je vhodné použít v případě, že zobrazujeme hodně malé území, nebo pokud je hlavním účelem měřit vzdálenosti (OpenGeo, 2012b).

Také bylo navrhováno, aby se všechny vrstvy, bod, linie i polygon, ukládaly do databáze do jedné tabulky. Databáze tohle umožňuje, typ prvku je vždy rozlišitelný prvním slovem v příkazu. Problém nastává u mapového serveru, který sice zobrazí všechny tři typy najednou, avšak nedokáže nadefinovat symbologii zvlášť. Mapfile totiž umožňuje nastavit TYPE pouze na anotaci, graf, kruh, bod, linii, polygon, rastr a dotaz. Takže když polygonu nastavím TYPE bod, vykreslí se mi jen body v místech, kde má polygon vrcholy.

Přestože prvotní myšlenka byla rozšířit framework pmapper, po rozboru bylo zjištěno, že do tak propracované aplikace by se dalo stěží zasahovat. Zvlášť pokud je z velké části postavena na php, což vzhledem k tomu, že se jedná o program fungující na straně serveru, nedovoluje tak rychlé odezvy při ovládání aplikace. Vhodnější se zdá například JavaScript, který pracuje na straně klienta a umožňuje tak rychlou obsluhu. S podobných technologií by se dal použít také například Ajax nebo Canvas, ale po objevení knihovny OpenLayers, která disponuje velkým množstvím tříd a metod pro tvorbu mapových aplikací, byla zvolena tato forma.

Další z prostředků bylo zvoleno PHP pro komunikaci mezi klientem a serverem částečně i proto, že bylo využito zkušeností s tímto jazykem. Jedná se o velice používaný jazyk právě pro účely správy databází a komunikaci mezi prohlížeči a servery.

Nejproblematictější část se ukázala být editace bodů, která může být atributová a geometrická. Funkčnost modulu je možná jen pokud jsou všechny programové prostředky správným způsobem nastaveny a kombinovány. Pro atributovou editaci byla využita metoda `WMSGetFeatureInfo()`, která pomocí webové mapové služby a dotazu `GetFeatureInfo` přenáší všechny hodnoty všech atributů pouze s výjimkou geometrie.

Pro přenos geometrie je určena služba WFS (Web Feature Services). Výsledkem tohoto přenosu je stránka ve formátu GML, která obsahuje veškeré údaje o prvcích včetně jejich souřadnic. Přestože byla nalezena metoda i způsob jak propojit jednotlivé technologie, nepovedlo se geometrickou editaci zcela zprovoznit.

Metoda knihovny OpenLayers, která umí zachytit přenos WFS, je `GetFeatureInfo()`, kde se definuje jako protokol url adresa CGI skriptu MapServer a cesta k Mapfile. Aby bylo možné poté s načteným GML pracovat, existuje metoda `addFeature()`, která prázdné vektorové vrstvě umí předat souřadnice ze souboru GML a překreslit je na vektory. Pak už by mělo jít použít stejných metod, jaké se používají při přidávání nového bodu.

Modul má mnoho možností k rozšíření s ohledem na způsob užívání konkrétním uživatelem. OpenLayers nabízí mnoho dalších ovládacích prvků, které zjednoduší práci při vytváření nových prvků. Vzhled je také možno upravit, nejlépe podle vzhledu aplikací používaných v rámci stejného projektu.

Práce především řeší způsob spojení jednotlivých technologií tak, aby vytvořily funkční celek. Ostatní záležitosti ohledně vzhledu či ovládání modulu už spíše závisí na konkrétním uživateli.

6 ZÁVĚR

V rámci bakalářské práce byl dle zadání navržen a vytvořen modul pro vstup a editaci dat pro mapový server University v Minnesotě. Plně funkční je editace atributů a byl navržen způsob, jakým by se měla řešit editace geometrie. Podmínkami bylo vytvořit modul na nekomerčním řešení, což bylo zajištěno programy s otevřeným zdrojovým kódem UMN MapServer, PHP, JavaScript, PostgreSQL. Bylo dodrženo, že se data budou ukládat dle standardů OGC, a byl použit i další standard WMS pro přenos dat.

Práce by nebyla možná bez důkladného nastudování všech technologií a jejich možností. V rámci rešerše byla nastudována problematika dostupnosti hotových řešení na internetu, která naznačila, že možnost editace je možná v nové verzi frameworku GeoMoose. Ostatní zdroje však většinou pouze ukazují možné řešení, nejsou však plně funkční ani nenabízí postup, jak aplikaci zprovoznit.

Hlavním výsledkem této bakalářské práce je funkční mapový modul, který se skládá z mapového pole a formuláře. Mapu lze interaktivně ovládat, tedy měnit měřítko, posouvat výřezy. Identifikace je možná po kliknutí na některý z prvků, pak se v popup okně zobrazí hodnoty prvků z databáze. Ve stejném okně je dále možno změnit některý z atributů či celý prvek smazat. Nový záznam do databáze lze vložit po nakreslení geoprvků a vyplnění polí ve formuláři pro vkládání nových prvků.

Tvorba modulu byla podrobně popsána v textové části, kde je důraz kladen na rozbor všech skriptů tak, aby podle nich bylo možno modul použít jiným počítačem. Bakalářská práce přináší řešení, které může být dále dle potřeb a požadavků vyvíjeno za účelem zpřístupnění dat více uživatelům v rámci počítačové sítě.

POUŽITÁ LITERATURA A INFORMAČNÍ ZDROJE

ALKOVÁ, Eva. Archeologický informační systém Olomouce. Olomouc, 2010. 56 s. Diplomová práce. Univerzita Palackého v Olomouci.

AULDS, Charles. Linux: administrace serveru Apache. 1. vyd. Praha: Grada, 2003, 535 s. ISBN 80-247-0640-7.

BRÁZA, Jiří. PHP 5: začínáme programovat. 1. vyd. Praha: Grada Publishing, 2005, 244 s. ISBN 80-247-1146-X.

JANÍKOVÁ, Věra. Informační systém rezervace Ščůrnica. Olomouc, 2010. Bakalářská práce. Univerzita Palackého v Olomouci.

MINÁŘ, Martin. Mapová aplikace pro zobrazování epidemiologických dat na webu. Brno, 2008. Bakalářská práce. Masarykova univerzita.

MITCHELL, Tyler. Web mapping illustrated. Sebastopol, Calif.: O'Reilly, 2005, 349 s. ISBN 978-059-6008-659.

OBE, Regina a Leo HSU. PostGIS in action. London: Pearson Education [distributor], 2011, 492 s. ISBN 19-351-8226-9.

PIETRASOVÁ, Petra. Možnosti tenkých a tlustých klientů při správě a analýze krajinně-typologických dat. Olomouc, 2008. Diplomová práce. Univerzita Palackého v Olomouci.

ŠIMÁČEK, Petr. Mapový server sadů, parků a zahrad města Olomouce. Olomouc, 2008. 75 s. Diplomová práce. Univerzita Palackého v Olomouci.

ŠREJBER, V. Komerční a nekomerční řešení distribuovaných GIS [dokument formátu pdf], 2002, [cit 2010-07-10].

Internetové zdroje

BEAUJARDIERE, Jeff de la. OpenGIS® Web Map Service (WMS) Implementation Specification [dokument formátu pdf]. 2006 [cit. 2012-08-08].

HERRRING, John R. OpenGIS® Implementation Standard for Geographic information - Simple feature access. [dokument formátu pdf]. 2010 [cit. 2012-08-08].

JÁCHYM, Čepický. Mapový server snadno a rychle (5). [online]. 2005 [cit. 2012-08-09]. Dostupné z: www.root.cz/clanky/mapovy-server-snadno-a-rychle-5/

JANOVSKÝ, Dušan. Jak psát web: Javascript - úvod [online]. 2012 [cit. 2012-08-08]. Dostupné z: <http://www.jakpsatweb.cz/javascript/javascript-uvod.html>

JIRÁNEK, Jan. WMS: Web Map Service [online]. 2008 [cit. 2012-02-12]. WMS typy dotazů (dle OGC). Dostupné z: <http://geo3.fsv.cvut.cz/wms/index.php?menu=wmsdotaz>

MapServer. MapServer 6.3.0 documentation – About. [online]. 2011 [cit. 2012-08-01]. Dostupné z: <http://mapserver.org/about.html>

OGC. OPEN GEOSPATIAL CONSORTIUM. About OGC [online]. 2012 [cit. 2012-08-08]. Dostupné z: <http://www.opengeospatial.org/og>

OPENGEO. Introduction to an Open Source Geostack [online]. 2012a [cit. 2012-08-08]. OpenLayers. Dostupné z: <http://workshops.opengeo.org/stack-intro/openlayers.html>

OPENGEO. Introduction to PostGIS [online]. 2012b [cit. 2012-08-08]. Section 17: Geography. Dostupné z: <http://workshops.opengeo.org/stack-intro/openlayers.html>

POSTGRESQL. FAQ - PostgreSQL wiki [online]. 2012 [cit. 2012-08-08]. Dostupné z: <http://wiki.postgresql.org/wiki/FAQ>

THE APACHE SOFTWARE FOUNDATION. About the Apache HTTP Server Project [online]. 2012 [cit. 2012-08-08]. Dostupné z: http://httpd.apache.org/ABOUT_APACHE.html

SUMMARY

The main aim of this bachelor thesis was to create a module for input and edition of data for mapserver. The module is built upon noncommercial software - the UMN Mapserver, PHP for communication with the database, PostgreSQL with the spatial extension PostGIS as a database system and JavaScript with the OpenLayers library for creating interactive map application. Data is stored according to the OGC standards and the map is provided by web map services (WMS) which also is part of the OGC standards.

Main contribution of this thesis is the ready-to-use map module which consists of the actual map field and a form. The map rendered in the map field can be interactively manipulated, e.g. the map scale can be changed and the currently displayed map segment can be shifted (panzoom feature). Identification of the geospatial data elements (when this feature is selected in the form) allows the user to view and edit the properties of the selected element in a pop-up window, that shows up after the desired element is selected in the map field, or even delete it completely. A new geodata element can be inserted into the database by drawing it in the map field and entering the necessary values in the form.

Broad study of the related matter and technologies was a necessity. It was found out there is no such free map application available, which allows the input and edition of data, on the internet, and there are not even any guidelines that would provide the means to solve this kind of problem.

The application (which forms the practical part of the thesis) was described in the text in enough detail that it should be possible to deploy it on another computer and to adapt it according to the requirements of particular deployment. The program was created as a web module to enable the edition of shared geospatial data in a computer network.

PŘÍLOHY

SEZNAM PŘÍLOH

Volné přílohy

Příloha 1 CD

Popis struktury CD

Adresáře a soubory:

Solanska_12 – obsahuje kompletní modul pro vstup a editaci dat

Solanska_12.sql – export geodatabáze ve formátu sql

web – webové stránky o bakalářské práci

Solanska_bp.pdf – text bakalářské práce