VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INTELLIGENT SYSTEMS

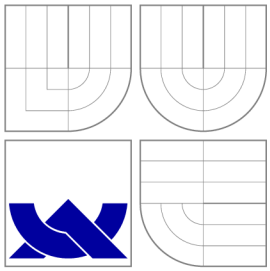
IMPLEMENTACE GIS NÁSTROJE PRO MOBILNÍ POČÍTAČOVÁ ZAŘÍZENÍ

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

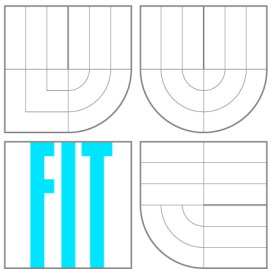
AUTOR PRÁCE
AUTHOR

Bc. TOMÁŠ PLACHÝ

BRNO 2015



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INTELLIGENT SYSTEMS

IMPLEMENTACE GIS NÁSTROJE PRO MOBILNÍ POČÍTAČOVÁ ZAŘÍZENÍ

GIS APPLICATION TOOL FOR MOBILE PLATFORMS

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE
AUTHOR

Bc. TOMÁŠ PLACHÝ

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. MARTIN HRUBÝ, Ph.D.

BRNO 2015

Abstrakt

Cíle této práce jsou návrh a implementace GIS nástroje pro mobilní počítačová zařízení. V úvodu práce shrnuje základní teorii okolo GISů a mapování. Následně je v práci rozebrán koncept mobilního mapovacího nástroje. Práce popisuje architekturu a procesy v rámci nástroje a abstraktního pohledu a uvádí některé implementační detaily. V rámci práce je provedeno mapování ve vybrané lokalitě a součástí práce je protokol shrnující provedené mapování.

Abstract

The aim of the thesis is to design and implement a GIS tool for mobile devices. The beginning of the thesis informs about the theoretical background of GIS and mapping. Subsequently, the thesis deals with the concept of a mobile mapping tool. The thesis describes the architecture and processes within the tool, as well as some details regarding its implementation. A mapping of a selected geographical location and a protocol summarising the mapping are also included within the thesis.

Klíčová slova

GIS, mapování, mapová vrstva, WFS, WMS, Spatialite, Shapefile, GPS, geozápisní.

Keywords

GIS, mapping, map layer, WFS, WMS, Spatialite, Shapefile,GPS, geodetic scrapbook.

Citace

Tomáš Plachý: Implementace GIS nástroje
pro mobilní počítačová zařízení, diplomová práce, Brno, FIT VUT v Brně, 2015

Implementace GIS nástroje pro mobilní počítačová zařízení

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením pana Ing. Martina Hrubého, Ph.D.

.....
Tomáš Plachý
27. května 2015

Poděkování

V rámci této příležitosti bych rád poděkoval panu Ing. Martinu Hrubému, Ph.D. za vedení a pomoc při vypracování této práce.

Obsah

Kapitola 1

Úvod

Tvorba geodatabází pro geografické informační systémy (dále jen GIS) je jednou z nejnáročnějších a nejdražších aktivit. Typicky 15-50% celkových výdajů za GIS je vynaloženo na získání dat. Pokud bychom do výdajů nepočítali výdaje za personál, pak se výdaje za data pohybují mezi 60-85% nákladů [13]. Jednou z možností pro získání geografických dat je přímé mapování v terénu. V této práci budeme pod pojmem mapování uvažovat dlouhodobou cílevědomou činnost, při které probíhá sběr geografických dat v terénu a vytváření objektů. Během mapování se geografická data zaznámenávají do geodetických zápisníků [12].

Tradiční zeměměřič určuje svou polohu pomocí triatelace. Za pomoci pruhované tyče a totální stanice změří svou vzdálenost od tří známých bodů, a z naměřených hodnot odvodí svou polohu. Přesnost tohoto měření je vysoká, ale vyžaduje předem sestavenou a udržovanou síť triangulačních bodů a proškoleného zeměměřiče.

Alternativním způsobem pro určení polohy je využití družicového navigačního systému GPS (Global Positioning System). GPS modul je v dnešní době součástí téměř každého mobilního zařízení a umožňuje tak snadné a rychlé získání polohy. Díky GPS jsme schopni určit svou polohu dostatečně přesně na to, abychom tyto údaje mohli využít pro sběr geografických dat.

Zde se nabízí příležitost využít možností mobilních zařízení a GPS pro vytvoření komplexního mapovacího nástroje. Nástroj, který by naměřená data přímo ukládal a umožňoval s nimi pracovat, by mohl značně zjednodušit a zrychlit průběh mapování.

1.1 Cíle práce

Cílem diplomové práce je návrh a implementace souboru programů mGIS v programovacím jazyce Java. mGIS slouží pro získávání geografických dat v terénu a skládá se ze dvou aplikací. Mobilní aplikace pro mapování mGISmobile pro operační systém Android a aplikace mGISserver pro pc. mGISserver slouží pro uložení a export naměřených dat. Komunikace mezi mGISmobile a mGISserver probíhá na základě standardu WFS pro přenos vektorových dat. Důraz je kladen na to, aby aplikace mGISmobile byla schopna pracovat i v offline režimu. Offline režim umožňuje provádět mapování i na místech, kde není k dispozici internetové pokrytí.

Kapitola 2

Geografický informační systém

Geografický informační systém [12, 14, 1] je elektronický informační systém, který umožňuje ukládat, spravovat, analyzovat a vizualizovat data, která mají prostorový vztah k povrchu Země. GIS pracuje s geodaty, která jsou definována svou geometrií, topologií a atributy. GIS není počítačový systém pro vytváření map, ale hlavně analytický nástroj pro práci s prostorovými vztahy mezi jednotlivými objekty. GIS, stejně jako obecný informační systém, se skládá z několika částí [1]:

- Hardware – počítač s možnostmi pro vstup a výstup dat (monitor, tiskárna, skener, GPS a další).
- Software – sada programů pro analýzu a vizualizaci dat.
- Data – nejdůležitější součást GISu.
- Uživatelé – lidé se znalostmi geografie a obsluhou informačních systémů.
- Metody – zapojení GISu do stávajícího IS podniku.

2.1 Modelování v GIS

Mapování geografického prostoru spočívá ve vytvoření modelu daného prostoru. Tvorba počítačového modelu je časově náročná, dle odhadů zabere 75-80% veškerého času [13]. Modelování zadaného problému prochází následujícími etapami [12]:

- Zkoumání reality a poznávání všech jejích aspektů.
- Formulování úlohy. Výsledkem této etapy by měl být jistý konceptuální model, kde zvýrazníme ty aspekty reality, které jsou relevantní pro naši úlohu.
- Realizace úlohy v prostředí GIS nástroje.
- Analýza údajů ve vytvořené GIS aplikaci. Snaha o dosažení vytyčeného cíle.

2.2 Prostor a geoobjekt

Definovat pojem prostor je mimořádně komplikované. Pojetí prostoru je velmi široké, sahá od fyzikálního až po abstraktní prostor, jenž je běžně definován jako množina prvků, která

má některé rysy reálného fyzikálního prostoru. Pro člověka je běžné pracovat s různými pojetími prostoru a přecházet mezi nimi, ovšem v prostředí informačních systémů je nutné pojem prostor formalizovat. Prostor může být koncipován dvěma způsoby (Worboys, 1995 [12]):

- Soubor poloh s definovanými vlastnostmi – absolutní prostor existující sám o sobě.
- Soubor objektů s prostorovými vlastnostmi – relativní prostor.

GIS pomocí mapových vrstev zavádí obě chápání prostoru. Absolutní prostor v podobě rastrové mapy, která má dané souřadnice okrajů a prostor rozděluje na malé plošky. A relativní prostor v podobě vektorové mapy, její umístění vyplývá z umístění objektů.

Kompletním popisem geoobjektu je jeho jednoznačná identifikace a jeho prostorová poloha. V mnoha aplikacích však údaj o poloze není pro práci s objektem dostačující. Objekt je tedy doplněn o další atributy, které ho blíže specifikují. Souhrnně je tedy objekt určen[12]:

- Identifikací – OID (object identificator), jednoznačně identifikuje objekt v rámci jedné vrstvy.
- Prostorovou polohou – geografická souřadnice, případně posloupnost souřadnic.
- Atributy – neprostorové údaje.
- Čas – čas zaznamenání (pouze v některých aplikacích).

2.3 Dimenze prostorových objektů

Podle potřeby geometrického modelování rozlišujeme různé prostorové dimenze objektů [12, 1, 14].

- Objekty bezrozměrné $0-D$ – body definované svou polohou.
- Objekty jednorozměrné $1-D$ – úseky čar s konečnou délkou, ale nulovou plochou.
- Objekty dvojrozměrné $2-D$ – polygony.
- Objekty trojrozměrné $3-D$ – polyhedrony.

Při mapování modelujeme objekty reálného světa, nemá tedy smysl zabývat se objekty s dimenzí vyšší než 3-D i když nám to matematika umožňuje.

Dimenze modelovaného objektu závisí na potřebách, pro které je vytvořen. Pokud potřebujeme pouze informaci o tom, že daný objekt se nálezá na daném místě, můžeme 3-D objekt zjednodušit až na 0-D objekt. Jako příklad vezměme jezero. Pokud potřebujeme pouze znázornit polohu jezera, vystačíme si s 0-D informací o jeho poloze. Pokud nás zajímá i rozloha nebo tvar jezera, pak ho budeme modelovat jako 2-D objekt. GIS modely se vztahují především k povrchu Země, a proto budeme ve většině případů 3-D objekty transformovat do 2-D.

2.4 Mapové vrstvy

Základní podobou geodatabázového souboru v GIS je mapová vrstva[12, 1, 14]. Vrstva v sobě slučuje tematicky podobné objekty, např. řeky, cesty, budovy a další. Mapovou vrstvu můžeme vnímat jako průhlednou fólii, na které je zobrazena část reality. Sloučením vrstev získáme požadovanou celkovou mapu. Rozdělení světa pomocí monotematických vrstev nám snadněji umožňuje zorganizovat a pochopit vztahy mezi jednotlivými jevy.

Vycházíme z poznatku, že vrstvy je snadné sloučit, ale komplikované je rozdělit. Oddělení různých typů vede také k většímu pořádku v datech. Jednotlivé soubory je možné přenášet mezi systémy, sdílet s ostatními uživateli a využívat jednotlivé vrstvy současně ve více projektech.

2.4.1 Vektorová vrstva

Vektorová vrstva se skládá z geoobjektů. Jak bylo řečeno v kapitole 2.2, každý geoobjekt je definován svou geometrií a jednoznačným identifikátorem OID. Zvyklostí je, že vektorové vrstvy jsou monotematické a všechny geoobjekty vrstvy mají stejný typ geometrie. Pak můžeme o vrstvách mluvit jako o bodových, liniových nebo polygonových.

Geometrie každého geoobjektu se skládá z bodů a linií. Bod má definovanou polohovou souřadnici, ale z geometrického hlediska nemá žádný rozměr. Linie je poté úsečka nebo křivka spojující dva body. Složitější geometrie, jako například polygony, se skládá z bodů a úseček.

Z hlediska datového modelu je uložení geometrie složité. Geometrie jednotlivých geoobjektů může být velmi složitá a v rámci jednoho systému ukládáme geometrie různých dimenzí. Na zvoleném způsobu uložení dat závisí také to, zda budou uloženy informace o topologických vztazích mezi jednotlivými geoobjekty. Pro uložení vektorových dat můžeme zvolit jeden z následujících vektorových modelů:

- Špagetový model – objekty jsou, bez ohledu na počet dimenzí, uloženy v jednom heterogenním seznamu. Seznam má 2 položky:
  - typ objektu – bod, linie, polygon.
  - parametry objektu – seznam souřadnic.

  Tento model neobsahuje žádné informace o topologii a je těžko použitelný pro analýzu dat.

- Hierarchický model – data jsou ukládána hierarchicky s ohledem na počet dimenzí. Vychází z faktů, že polygon se skládá z několika linií, linie se skládá z několika úseček a úsečka je spojení dvou bodů. Jednotlivé elementy jsou uloženy samostatně a model obvykle obsahuje topologické informace. Oproti topologickému modelu umožňuje při vyhledávání použít jen část datových struktur.

- Topologický model – jeden z nejpoužívanějších modelů uchovávající topologické informace. Linie začíná a končí v bodě (uzlu), které jsou uloženy jako soubor souřadnic. Každá část linie je uložena s odkazem na její okrajové uzly. Dvě linie se mohou protínat pouze v uzlu. Ve struktuře jsou u jednotlivých linií uloženy informace o pravém a levém polygonu, které zachovávají základní prostorové vztahy pro analýzu.

Datový model musí umožňovat i uložení atributů pro jednotlivé geoobjekty. Tradičním způsobem je uložení geometrie a atributů odděleně. Možným řešením je v takovém případě

takzvaná atributová tabulka. Atributová tabulka obsahuje názvy a hodnoty jednotlivých atributů, společně s OID geoobjektu. Pomocí OID lze následně spárovat geoobjekt a jeho atributy.

2.4.2 Rastrová vrstva

Rastrová vrstva [12, 1, 14] reprezentuje výřez prostoru. Používá se k modelování veličin, které jsou spojitě definovány na celé prostoru, např. nadmořská výška. V rastrové vrstvě je prostor dělen na malé plochy,r může být dělen pravidelně nebo nepravidelně. Buňky mohou být různého tvaru (čtverec, trojúhelník, šestiúhelník). V naprosté většině případů se používá čtvercová mřížka. Rastrové vrstvy můžeme dělit do několika typů:

- Rastrové vrstvy výčtového typu – každá buňka obsahuje kód, typicky hodnota z rozsahu 1..n. Kód reprezentuje kategorii sledovaného jevu a součástí vrstvy je tabulka, podle které se kódy interpretují. Používají se tam, kde existuje konečný počet hodnot, nebo lze veličinu rozdělit do konečného počtu kategorií, např. typ vegetace.
- Rastrové vrstvy hodnotového typu – každá buňka obsahuje informaci o diskretizované hodnotě spojité veličiny, např. nadmořská výška, teplota.
- Podkladové vrstvy – používají se jako podklad při práci s vektorovými vrstvami. Běžně se jedná o fotografie z dálkového průzkumu Země nebo rendrované mapy.

2.4.3 Zdroje podkladových vrstev

V GISech běžně využíváme rastrové vrstvy právě jako podklad pro práci s vektorovými objekty. Aplikace mohou pracovat s předem připravenými a dodanými daty ve formě souborů nebo data získávat online. V následujích sekcích jsou popsány některé z dostupných zdrojů podkladových vrstev a jejich využitelnost na mobilních zařízeních.

Google Maps Android API

Google Maps API [22] je integrováno přímo v sytému Android a umožňuje v aplikaci zobrazovat data z Google Maps. API poskytuje komponentu mapView pro zobrazení mapy a umožňuje do mapy zakreslovat vlastní body, čáry a polygony. Komponenta mapView vývojáře odstiňuje od vykreslování mapy, načítání dat mapy, a poskytuje tak jednoduchý způsob pro integraci map do vlastní aplikace.

Google Maps API neposkytuje možnost uložení map pro offline použití. Toto omezení je hlavní překážkou pro využití Google Maps v této práci. Navíc ukládání informací z Google Maps je striktně omezeno v licenčních podmínkách [23]. Omezení ohledně offline použití dat je hlavní překážkou pro použití Google Map v implementované aplikaci i když poskytované mapy dosahují vysoké kvality.

OpenStreetMap

OpenStreetMap (dále jen OSM) je vybudována komunitou, která se stará o udržování a doplňování obsažených informací. Data z OSM jsou poskytována volně k využití pod licencí ODbL[31].

OSM je možné na platformě android použít prostřednictvím knihovny OSMDroid (kapitola 2.12.4). OSMDroid knihovna poskytuje komponentu mapView podobnou komponentě

poskytované Google Maps API (kapitola 2.4.3). Komponentu lze jednoduše zakomponovat do vytvářené aplikace a pro zobrazení mapy od programátora vyžaduje pouze nastavení zdroje mapových dlaždic.

Alternativou k zobrazování dat pomocí knihovny je využít přímo mapové dlaždice z dlaždicových serverů [41]. Dlaždice[41] jsou čtvercové výřezy mapy, typicky o velikosti 256x256 pixelů. Mapa světa je rozdělena čtvercovou mřížkou na jednotlivé dlaždice. Velikost oblasti, která je obsažena v jedné dlaždici, je dáná úrovní přiblížení. Na úrovni 1 je mapa světa rozdělena do čtyř dlaždic a s každou další úrovní dochází k dělení jednotlivých dlaždic na čtvrtiny. Každá dlaždice je tedy určena svou souřadnicí v mřížce a úrovní přiblížení. Při přímém využití mapových dlaždic musí programátor implementovat výpočet souřadnice požadované dlaždice dle zvolené úrovně přiblížení, stažení příslušné dlaždice ze serveru například pomocí WMS 2.8 a vykreslení dlaždice na obrazovku zařízení.

Georeferencovaný rast

TIFF [10] je univerzální široce používaným datový formát pro ukládání rastrových dat. TIFF byl používán pro ukládání satelitních snímků a vznikla potřeba vložení geografických dat do souboru pro jednoduché využití v geografických systémech. Vznikl veřejný standard formátu GeoTIFF [7], což je TIFF soubor s uloženými geografickými daty ve formě tagů. Při zobrazení se geografická data mohou použít pro umístění na správné souřadnice.

Použití GeoTIFF souborů jako podkladové mapy by vyžadovalo přípravu podkladů před každým mapováním. Uživatel by si musel připravit podklady pro zvolenou oblast ve všech potřebných úrovních přiblížení. Následně připravené podklady nahrát do mobilního zařízení. V aplikaci by nebyla možnost stahování mapových podkladů v terénu pomocí internetového připojení a uživateli by se mohlo stát, že se dostane do oblasti, kde nebude mít podkladové mapy k dipozici. Z implementačního hlediska by přibyla nutnost zpracování geografických dat uložených v souborech, a na jejich základě správné vykreslování souborů v aplikaci.

2.5 Referenční elipsoid

Evolucí vznikl zvrásněný a členitý povrch Země. Povrch Země se velmi liší od povrchu koule nebo rotačního elipsoidu, které jsou matematicky jednoduše popsatelné. Protože každý bod Země může být předmětem zkoumání, potřebujeme povrch Země zjednodušit, aby bylo možné mapování reálného povrchu Země na plochu mapy.

Počáteční aproximací je takzvaný geoid[12]. Je definován jako plocha se stejnou hodnotou zemské tíže v každém jejím bodě. Tato plocha je položena na úroveň klidné střední hladiny moří, tím pádem zasahuje na některých místech pod povrch pevniny.

Geoid vyhlazuje zvrásněný povrch Země, ale stále není jednoduše matematicky popsatelný. Zavádí se další model geoidu – tím je náhradní elipsoid. Jde o rotační elipsoid, který je dán svým středem a délkami poloos. Různé parametry elipsoidu nahrazují geoid s různou přesností, protože se na různých místech mění vzdálenost plochy geoidu od elipsoidu. Povrch geoidu vůči referenčnímu elipsoidu se může lišit až o 100 metrů.

Pro účely velkých mapových děl je nezbytné data vztahovat ke globálnímu elipsoidu. V roce 1984 se ustálil globální elipsoid v podobě zvané WGS-84 (World Geodetic System 1984). Tento elipsoid se používá např. při satelitní navigaci GPS. Ostatní elipsoidy lze definovat pomocí posunutí jejich středu, případně změnou délky poloos, vůči WGS-84[45].

2.6 Geografické souřadné systémy

Souřadný systém[12] je nástroj k vyjádření polohy v nějakém prostoru. V případě GISů mluvíme o geografickém prostoru. Znalost souřadného systému je pro GIS nezbytnou podmínkou pro transformaci souřadnic mezi systémy. Souřadné systémy můžeme rozdělit na dva typy:

- Globální – snahou je postihnout celý geografický prostor Země. Výhodou je univerzálnost v popisu planet a přenositelnost do jiných GIS systémů. Prostor globálního souřadného systému je povrch elipsoidu.
- Lokální – vytvořené pro popis menšího území. Tyto systémy umožňují pro výpočet vzdálenosti mezi body využít Eukleidovskou metriku.

2.6.1 Systém šířka-délka (WGS-84)

WGS-84 [45, 12] je světově uznávaný souřadný systém vydaný ministerstvem obrany USA v roce 1984. Referenčním systémem elipsoidem je elipsoid WGS-84. Tento souřadný systém je využívám například systémem GPS. Souřadnice bodu se skládá ze zeměpisné šířky a délky.

Zeměpisná šířka nabývá hodnot a 0-90° a hodnota je úhel, který svírá bod na povrchu Země s rovinou rovníku. Rovníku je přiřazena hodnota 0° a rozděluje Zemi na dvě polokoule. Severní pól má 90° severní šířky, obdobně jižní pól má 90° jižní šířky. Při počítačovém vyjádření zeměpisné šířky se používají kladné hodnoty pro severní, a záporné pro jižní polokouli.

Zeměpisná délka nabývá hodnot 0-180°. Obdobně jako u zeměpisné šířky, tato hodnota je dána úhlem, který bod svírá s rovinou nultého poledníku. Na rozdíl od rovníku, který je přirozenou rovinou, neexistuje přirozená rovina rozdělující Zemi na východní a západní polokouli, což se v historii projevilo problémy při určování polohy. Dohodou byl jako nultý poledník stanoven poledník procházející hvězdárnou v Greenwich. Pro vyjádření východní délky používáme kladné hodnoty, záporné hodnoty pak pro západní délku.

2.7 Mapové zobrazení

Mapové nebo také kartografické zobrazení [11] je způsob, kterým se převádí zobrazení povrchu Země nebo jeho části do dvojrozměrné roviny mapy. Každému bodu na povrchu Země je přiřazen bod v rovině mapy. Každé zobrazení s sebou nese určité zkreslení a výběr vhodného zobrazení závisí na účelu výsledné mapy.

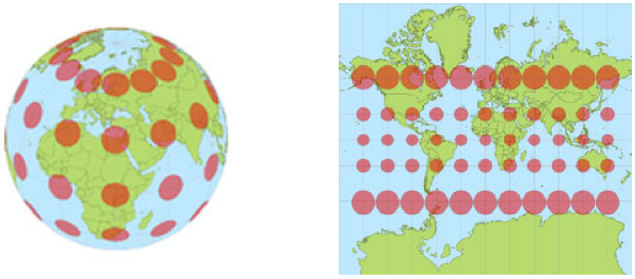
Mapová zobrazení může rozdělit do základních skupin podle následujících kritérií:

- Kartografické zkreslení – podle toho, zda zobrazení nezkresluje plochy, úhly nebo délky.
- Zobrazovací plocha – na jakou plochu povrch zobrazujeme (rovina, plášť kužele, plášť válce).
- Poloha osy zobrazovací plochy – orientace osy válce nebo kužele k ose glóbu.

2.7.1 Mercatorovo zobrazení

Mercatorovo zobrazení je druh úhlojevného válcového mapového zobrazení. Toto zobrazení se používá zejména v navigačních mapách. OpenStreetMap, které jsou v rámci diplomové práce využity jako podkladová vrstva, využívají právě Mercatorovo zobrazení.

Mercatorovo zobrazení zobrazuje zemský povrch na plášť válce dotýkajícího se glóbu na rovníku, jehož osa je rovnoběžná s osou Země. Úhlojevné zobrazení zachovává úhly, ale zkresluje plochy. Na obrázku 2.1 je vidět výsledek zobrazení plochy Země a vzniklé zkreslení. Po zobrazení vzniká pravoúhlá síť poledníků a rovnoběžek, kde poledníky mají mezi sebou stejné rozestupy, ale rozestupy rovnoběžek se směrem k pólům zvětšují. Kvůli zvětšujícímu se zkreslení směrem k pólům je Mercatorovo zobrazení nepoužitelné při tvorbě map polárních oblastí.



Obrázek 2.1: Mercatorovo zobrazení s vyznačeným zkreslením [29] .

2.8 Web Map Service

Web Map Sevice[43] (dále jen WMS) je služba pro sdílení geografických informací v internetu ve formě rastru. Jedná se o standard vyvinutý a dále rozšiřovaný Open Geospatial Consortium (OGC). Výsledkem dotazu na WMS server jsou primárně obrazová data zobrazující informace z jedné nebo více mapových vrstev. Výsledný obrázek je georeferencován, má jednoznačně daný referenční souřadný systém a souřadnicový obdélník v tomto systému, který výsledná data ohraničuje.

2.8.1 Princip funkce

Aplikace pracuje na principu klient-server. Server obsahuje nastavení s možnostmi serveru, georeferencovaná data (GeoTIFF, SHP, aj.) a informace o geografických objektech. Klient komunikuje se serverem pomocí HTTP (Hyper Text Transfer Protocol) dotazů GET a POST. Na základě požadavku server zpřístupní klientovi obrazová data, která klient prezentuje uživateli.

2.8.2 Operace WMS

Klient komunikuje s mapovým serverem prostřednictvím třech základních operací [43, 44]:

- GetCapabilities – Odpovědí je XML soubor popisující danou službu. Soubor obsahuje metadata mapového serveru.
- GetFeatureInfo – Odpovědí je XML soubor s atributy daného prvku na mapě.

- GetMap – Tento dotaz lze považovat za hlavní. Odpovědí je mapa ve formě obrazových dat v určitém formátu.

2.8.3 Výhody a nevýhody služby

Specifikace služby s sebou přináší výhody i nevýhody, se kterými při jejím používání musíme počítat [28].

Výhody služby:

- WMS server umožňuje uživateli získat jen data (vrstvu), která skutečně potřebuje.
- Uživatel není závislý na typu mapového serveru a operačním systému.
- Uživatel může k datům přistupovat z různých hardwarových zařízení s připojením k internetu.
- WMS je navržena tak, aby podporovala nejrůznější souřadnicové systémy.

Nevýhody služby:

- Zařízení musí mít přístup na internetové připojení.
- Některé typy serverů se plně neřídí specifikacemi OGC.
- Klient získá pouze obrazová data.

2.9 Web Feature Service

Web Feature Service [42] (dále jen WFS) je služba pro poskytování geografické informace v internetu ve formě vektorových dat. Jedná se o standard vyvinutý a dále rozšiřovaný Open Geospatial Consortium (OGC). Výsledkem dotazu na WFS server jsou primárně geodata ve formátu GML [6]. Výsledná data jsou vztažena k referenčnímu souřadnému systému. V základu je služba podobná WMS.

2.9.1 Princip funkce

Aplikace pracuje na principu klient-server. Server obsahuje nastavení s možnostmi serveru a georeferencovaná vektorová data. Klient komunikuje se serverem pomocí HTTP (Hyper Text Transfer Protocol) dotazů GET a POST. Na základě požadavku server zpřístupní klientovi data ve formátu GML, která jsou po zpracování prezentována uživateli.

Základní typy WFS a operace, které musí poskytovat:

- Basic WFS – dotazy GetCapabilities, GetFeature a DescribeFeatureType.
- Xlink WFS – Basic WFS + dotaz GetGmlObject.
- Transactional WFS – Basic WFS + dotazy InsertFeature, UpdateFeature, DeleteFeature, ReplaceFeature.

2.9.2 Operace Transactional WFS

Transactional WFS poskytuje operace definované pro službu Basic, sloužící k získání dat dostupných na serveru. A přidává k nim operace pro manipulaci s objekty na serveru. Následující sekce popisují funkci a strukturu zpáv jednotlivých operací.

GetCapabilities

Tato operace je prováděna při zahájení komunikace se serverem. Dotaz slouží k získání základních informací o službě, adres pro zasílání požadavků pro jednotlivé operace a informace o dostupných mapových vrstvách.

Na obrázku 2.2 vidíme ukázku dotazu. Struktura dotazu je jednoduchá, obsahuje pouze název požadované operace, název služby a verzi. Název služby je velmi důležitý, protože struktura požadavku je stejná jako u služby WMS (kapitola 2.8).

```
<GetCapabilities
  service="WFS"
  version="1.0.0"
  xmlns="http://www.opengis.net/wfs"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"/>
```

Obrázek 2.2: GetCapabilities dotaz.

Odpověď (obrázek 2.3) můžeme rozděli na tři hlavní části. Část `Service` obsahuje základní informace o službě. V části `Capability` je uveden seznam všech operací, které server poskytuje. U každé operace je pak uveden typ návratové hodnoty a adresa pro zaslání požadavku. Poslední částí je `FetureTypeList`, která obsahuje seznam dostupných operací a mapových vrstev.

DescribeFeatureType

Operace `DescribeFeatureType` slouží k získání struktury jednotlivých záznámů vrstvy. Tato informace je pro aplikaci nutná pro správnou interpretaci dat jednotlivých geoobjektů vrstvy.

Na obrázku 2.4 vidíme strukturu požadavku. Základem dotazu je element `DescribeFeatureType` s atributy určujícími typ a verzi služby. Uvnitř se nachází seznam jmen mapových vrstev, pro které požadujeme strukturu dat.

Odpovědí na dotaz je `XML schema`, definující strukturu xml záznamu objektu vrstvy. Schema obsahuje jeden element pro každou mapovou vrstvu obsaženou v dotazu. Pro kadžý element vrstvy je v odpovědi definovaný vlastní datový typ. V rámci definice datového typu je uveden typ geometrie, názvy a typy jednotlivých atributů. Na obrázku 2.5 vidíme ukázku odpovědi pro dotaz na mapovou vrstvu obsahující geometrii typu polygon a jeden textový atribut.

GetFeature

Operace `GetFeature` slouží k získání dat jednotlivých geoobjektů vrstvy. Součástí dotazu mohou být kritéria pro omezení získaných objektů, například získání objektů v uvedeném bounding boxu.

Základem dotazu je element `GetFeature` s atributy určujícími typ a verzi služby. Uvnitř se nachází seznam elementů `Querry`, kde pro každou požadovanou vrstvu, existuje v seznamu právě jeden element. Každý element obsahuje atribut `typeName`, jehož hodnota odpovídá názvu vrstvy. Na obrázku 2.6 je vidět dotaz na objekty vrstvy s názvem `domy`.

Odpověď obsahuje seznam elementů `featureMember`. Každý element seznamu obsahuje data jednoho geoobjektu. Struktura dat geoobjektu je totožná se strukturou definovanou

```
<WFS_Capabilities version="1.0.0">
  <Service>
    <Name>GISDIP</Name>
    <Title>GIS pro mobilni zarizeni</Title>
    <Abstract>WFS pro diplomovou praci</Abstract>
    <Fees>none</Fees>
    <AccessConstraints>none</AccessConstraints>
  </Service>
  <Capability>
    <Request>
      <GetFeature>
        <ResultFormat><GML2 /></ResultFormat>
        <DCPType><HTTP>
          <Post onlineResource="http://127.0.0.1:8080/wfs" />
        </DCPType></HTTP>
      </GetFeature>
    </Request>
  </Capability>
  <FeatureTypeList>
    <Operations>
      <Insert />
      <Update />
    </Operations>
    <FeatureType>
      <Name>domy</Name>
      <SRS>EPSG:4326</SRS>
    </FeatureType>
  </FeatureTypeList>
```

Obrázek 2.3: GetCapabilities odpověď.

```
<DescribeFeatureType xmlns="http://www.opengis.net/wfs"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://www.opengis.net/wfs../wfs/1.0.0/WFS-basic.xsd"
   version="1.0.0"
   service="WFS">
   <TypeName>domy</TypeName>
</DescribeFeatureType>
```

Obrázek 2.4: DescribeFeatureType dotaz.

v odpovědi na dotaz `DescribeFeatureType` (kapitola 2.9.2). Na obrázku 2.7 vidíme ukázku odpovědi obsahující element typu `dum`. Struktura elementu `dum` odpovídá struktuře schematu na obrázku 2.5.

Transaction

Jako transakce jsou v rámci služby WFS označovány operace insertFeature, updateFeature, deleteFeature a lockFeature. V požadavku na transakci je vždy kořenovým elementem `Transaction`. Uvnitř se mohou nacházet následující elementy:

```
<xsd:schema
  xmlns:wfs="http://www.opengis.net/wfs"
  xmlns:gml="http://www.opengis.net/gml"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  version="1.0.0" >
  <xsd:import namespace="http://www.opengis.net/gml"
    schemaLocation=
      " http://http://schemas.opengis.net/gml/2.1.2/feature.xsd" />
  <xsd:element name="dum" type="dumType"
    substitutionGroup="gml:_Feature" />
  <xsd:complexType name="dumType">
    <xsd:complexContent>
      <xsd:extension base="gml:AbstractFeatureType">
        <xsd:sequence>
          <xsd:element name="barva" type="xsd:string" />
          <xsd:element name="geom" type="gml:PolygonMemberType"/>
        </xsd:sequence>
      </xsd:extension>
    </xsd:complexContent>
  </xsd:complexType>
</xsd:schema>
```

Obrázek 2.5: DescribeFeatureType odpověď.

```
<GetFeature
  xmlns="http://www.opengis.net/wfs"
  xmlns:gml="http://www.opengis.net/gml"
  version="1.0.0"
  service="WFS">
  <Query typeName="domy" />
</GetFeature>
```

Obrázek 2.6: GetFeature dotaz.

- Insert – vložení nového geoobjektu. Element obsahuje data vkládaného objektu. Struktura dat musí odpovídat struktuře definované v odpovědi na operaci `DescribeFeatureType`.
- Update – úprava existujícího geoobjektu. Atributem elementu je název vrstvy. Uvnitř elementu se nachází element `property` s novými daty a `filter` s podmínkami pro výběr objektu. Na obrázku 2.8 vidíme ukázku požadavku na update geoobjektu.
- Delete – smazání geoobjektu. Atributem elementu je název vstvy. Uvnitř se nachází element `filter` s podmínkami pro výběr objektu.
- Replace – nahrazení existujícího geoobjektu jiným. Atributem elementu je název vstvy. Uvnitře elementu se nachází data nového objektu.

Jak můžeme vidět na obrázku 2.9, odpověď na transakci obsahuje přehled jednotlivých operací. U každé operace je uveden počet geoobjektů, které byli v rámci transakce ovlivněny.

```
<FeatureCollection
   xmlns="http://www.opengis.net/wfs"
   xmlns:gml="http://www.opengis.net/gml" >
   <gml:featureMember>
      <dum>
         <barva>modra</barva>
         <geom>
            <gml:Polygon>
               <gml:outerBoundaryIs>
                  <gml:LinearRing>
                     <gml:coordinates decimal="." cs="," ts=" ">
                        3.0,3.0 6.0,3.0 6.0,5.0 4.0,5.0 3.0,3.0
                     </gml:coordinates>
                  </gml:LinearRing>
               </gml:outerBoundaryIs>
            </gml:Polygon>
         </geom>
      <dum>
   </gml:featureMember>
</FeatureCollection>
```

Obrázek 2.7: GetFeature odpověď.

```
<Transaction xmlns="http://www.opengis.net/wfs">
   <Update typeName=''domy''>
      <Property>
         <barva>modra</barva>
      </Property>
      <Filter>
         <Id>56</Id>
      </Filter>
   </Update>
</Transaction>
```

Obrázek 2.8: Požadavek na update objektu.

```
<TransactionResponse xmlns="http://www.opengis.net/wfs">
   <TransactionSummary>
      <totalInserted>0</totalInserted>
      <totalUpdated>3</totalUpdated>
      <totalReplaced>0</totalReplaced>
      <totalDeleted>0</totalDeleted>
   </TransactionSummary>
</TransactionResponse>
```

Obrázek 2.9: Odpověď po zpracování transakce.

2.10 Shapefile

Shapefile[5] je formát datového souboru pro uložení vektorových prostorových dat vytvořeného firmou Esri. Formát byl poprvé představen na začátku 90. let 19. století [20].

Shapefile umožňuje ukládat vektorová data ve formě bodů, čar a polygonů. Ke každému vektorovému objektu je možné uložit atributy, které ho popisují. Výhodou je jednoduchost formátu. Formát shapefile se skládá z několika souborů se stejným názvem uložených v té samé složce. Povinné jsou soubory s příponami .shp, .shx, .dbf. Záznamy v souborech .shx a .dbf musí odpovídat pořadí záznamů v souboru .shp.

2.10.1 Hlavní soubor (.shp)

Hlavní soubor[5] obsahujuje geometrická data v binární podobě. Soubor začíná hlavičkou fixní délky následovanou jednotlivými záznamy proměnlivé délky skládající se z hlavičky fixní délky a dat. Shapefile umožňuje ukládat 2D, 3D a 4D objekty.

Tabulka 2.1 popisuje strukturu hlavičky souboru. Hlavička má délku 100 bytů, obsahuje délku souboru, typ uložených objektů a minimální ohraničující čtverec objektů uložených v souboru. Pokud je soubor prázdný, hodnoty Xmin, Ymin, Xmax, Ymax nejsou specifikované.

Podporované typy objektů jsou vypsány v tabulce 2.2. Nespecifikované hodnoty jsou rezervovány pro budoucí použití. V současnosti musí být všechny objekty v shapefile stejného typu, pokud budou v budoucnu implementovány smíšené typy, dojde k zavedení nového typu pro jejich reprezentaci.

Hlavička jednotlivých objektů má délku 8 bytů a skládá se z čísla záznamu a délky dat (Tabulka 2.3). Za hlavičkou následují data lišící se dle typu objektu.

Objekt typu 0 je objekt bez geometrických dat. Všechny typy objektů podporují null hodnotu, je tedy možné mít v souboru s typem 1 (Point) i objekty typu 0 (Null Shape). Záznam objektu typu 0 je pouze první část záznamu udávající typ objektu (2.4). Tabulka 2.5 ukazuje strukturu záznamu pro bod. Záznam obsahuje typ objektu a souřadnice v případě dvourozměrného objektu X a Y. U složitějších objektů jako je polygon (Tabulka 2.6) obsahuje záznam kromě typu a souřadnic jednotlivých bodů i ohraničující čtverec, počet bodů a počet částí. Přehled struktury záznamů pro ostatní typy objektů je možné nalézt v dokumentaci.

Pozice	Hodnota	Typ	Pořadí Bytů	Popis
0	9994	Integer	Big-endian	Kód souboru
4-20	0	Integer	Big-endian	Nepoužité
24	Délka soboru	Integer	Big-endian	Délka soboru
28	1000	Integer	Little-endian	Verze
32	Typ objektu	Integer	Little-endian	Typ objektu
36-60	Bounding Box	Double	Little-endian	Xmin,Ymin,Xmax,Ymax
68-76	Bounding Box	Double	Little-endian	Zmin,Zmax - 0.0 pokud se nepoužívá
84-92	Bounding Box	Double	Little-endian	Mmin,Mmax - 0.0 pokud se nepoužívá

Tabulka 2.1: Struktura hlavičky souboru.

Hodnota	Typ objektu
0	Null Shape
1	Point
3	PolyLine
5	Polygon
8	MultiPoint
11	PointZ
13	PolyLineZ
15	PolygonZ
18	MultiPointZ
21	PointM
23	PolyLineM
25	PolygonM
28	MultiPointM
31	MultiPatch

Tabulka 2.2: Typy prostorových objektů.

Pozice	Hodnota	Typ	Pořadí Bytů	Popis
0	Číslo záznamu	Integer	Big-endian	Číslo záznamu - začíná číslem 1
4	Délka dat	Integer	Big-endian	Délka dat

Tabulka 2.3: Struktura hlavičky záznamu.

Pozice	Hodnota	Typ	Pořadí Bytů	Popis
0	Číslo záznamu	Integer	Big-endian	Číslo záznamu - začíná číslem 1

Tabulka 2.4: Struktura hlavičky null objektu.

Pozice	Hodnota	Typ	Pořadí Bytů	Popis
0	1	Integer	Little-endian	Typ objektu
4	X	Double	Little-endian	Souřadnice X
12	Y	Double	Little-endian	Souřadnice Y

Tabulka 2.5: Struktura dat bodu.

Pozice	Hodnota	Typ	Pořadí Bytů	Popis
0	3	Integer	Little-endian	Typ objektu
4	Box	Double	Little-endian	Ohraničující čtverec
36	Počet částí	Integer	Little-endian	Počet částí
40	Počet bodů	Integer	Little-endian	Počet bodů
44	Části	Integer	Little-endian	Části
X	Body	Point	Little-endian	Body: X = 44 + 4 * Počet částí

Tabulka 2.6: Struktura dat polygonu.

2.10.2 Indexový soubor (.shx)

Indexový soubor[5] je možné využít k rychlému vyhledávání vpřed i vzad. Soubor obsahuje stejnou hlavičku jako hlavní soubor (Tabulka 2.1), následovanou záznamy fixní délky 8 bytů, které obsahují offset příslušného záznamu v hlavním souboru a jeho délku (Tabulka 2.7). Délka dat je shodná s délkou uloženou v hlavním souboru.

Pozice	Hodnota	Typ	Pořadí Bytů	Popis
0	Offset	Integer	Big-endian	Offset v 16-ti bytových slovech od začátku souboru
4	Délka dat	Integer	Big-endian	Délka dat

Tabulka 2.7: Struktura záznamu indexového souboru.

2.10.3 dBASE soubor (.dbf)

Jedná se o binární soubor ve formátu dBASE[4, 5], ve kterém jsou uloženy atributy jednotlivých objektů. Soubor se skládá z hlavičky a datových záznamů. Hlavička začíná na bytu 0 a obsahuje definici struktury tabulky. Ukončením hlavičky je hodnotou 0x0D a za ní následují jednotlivé záznamy. Kompletní struktura souboru je uvedena v dokumentaci .

Soubor musí splňovat následující podmínky[5]:

- Tabulka musí obsahovat jeden záznam pro objekt.
- Pořadí záznamů musí být shodné s pořadím v hlavním souboru.
- Hodnota roku v hlavičce souboru musí být rok od roku 1900 (v případě roku 2015 bude v hlavičce zapsán rok 115).

2.10.4 Omezení

Při používání shapefile formátu se setkáme s několika omezeními při ukládání geometrie, které vycházejí ze standardu shapefile formátu nebo s omezeními atributů, které vycházejí ze standardu dBase.

Omezení shapefile jsou následující[21]:

- Není možné uložit topologické informace.

- Maximální velikost jednoho souboru je 2 GB (asi 70 milionů bodů).
- Shapefile nepodporuje křivky.
- Shapefile fyzicky umožňuje uložení různých typů objektů v jednom souboru, ale dokumentace uvádí, že všechny objekty, které nejsou typu 0, musí být stejného typu. Kombinace typu je dle dokumentace tedy limitována na kombinaci jednoho typu s typem 0.

Omezení atributů uložených v shapefile[21]:

- Dle standardu dBase je v názvech polí podporována pouze znaková sada ANSI.
- Názvy polí mohou mít maximálně 10 znaků.
- Maximální množství polí je 255.
- Čísla jsou uložená v textovém formátu, což může vést k chybám v reálných číslech.

2.11 Existující mobilní nástroje

Mobilní aplikace pro sběr geografických dat není žádnou novinkou a na trhu již existuje několik jejich zástupců. Od základní aplikace umožňující pouhé prohlížení dat až po aplikace určené pro prefesionály a speciální zařízení nabízející pokročilé funkce. Následující sekce představují několik zástupců existujících aplikací, jejich funkce a případná omezení v porovnání s aplikaci vytvořenou v rámci této práce.

2.11.1 ArcPad

ArcPad[19] je mobilní aplikace pro mapování a sběr dat v terénu vytvořená firmou Esri. Aplikace je vytvořena pro GIS profesionály v malých až středních týmech a zahrnuje pokročilé funkčnosti GIS pro editaci a zobrazování geografických informací. Aplikace je schopná pracovat v offline režimu a lze ji používat i na přístrojích bez GPS, v takovém případě probíhá tvorba prostorových dat prostřednictvím klikání na mapě. Uživatelské rozhraní není přizpůsobeno mobilním zařízením a malá tlačítka nutí k ovládání stylusem. Aplikace je dostupná pro systémy: Windows Mobile 5.0, 6.0, 6.1, 6.5 a Windows 8 tablety. Využívání aplikace je podmíněno zakoupením licence.

V porovnání s mGIS je ArcPad určen pro profesionály a obsahuje pokročilé funkce pro editaci geografických dat, ale zaostává nepohodlným ovládání, které není přizpůsobeno pro ovládání prsty.

2.11.2 ArcGIS for Windows Mobile

ArcGIS for Windows Mobile[18] od firmy Esri je určen k poskytování GIS funkcí a dat z centralizovaných serverů na mobilních zařízeních. Aplikace je navržena tak, aby ji mohli používat profesionálové v různě velkých organizacích bez rozsáhlých znalostí GIS. Uživatelské rozhraní je podobné jako v aplikaci ArcPad a nutí k ovládání stylusem. Aplikace je dostupná pro systémy: Windows Mobile 6.0, 6.1, 6.5 a Windows Embedded Handheld 6.5. Licence je součástí licenčních balíčků na produkty ArcGIS.

Hlavními nevýhodami je nepřizpůsobení ovládání aplikace pomocí prstů a nemožnost práce v offline režimu.

2.11.3 ArcGIS app for smartphones and tablets

ArcGIS[17] je mobilní aplikace od firmy Esri pro operační systémy Android, iOS a Windows Phone. Funkce aplikace jsou podobné jako u předchozí ArcGIS for Windows Mobile. Uživatelské rozhraní je již přizpůsobeno k ovládání prsty a aplikace je k dispozici zdarma. Aplikace nepodporuje offline režim a pro práci s ní je tedy zapotřebí ArcGIS Server, pro který je nutné zakoupit licenci. Aplikace je dostupná pro operační systémy: Android 2.3.3 a vyšší, iOS 5 a vyšší, Windows Phone 7.5 a vyšší.

Nevýhodou oproti mGIS je nutnost stálého připojení k internetu.

2.11.4 SHP Viewer a Shapefile over Map

SHP Viewer[36] a Shapefile over Map[35] jsou zdarma dostupné aplikace pro android, které umožňují prohlížení obsahu shapefile souborů (kapitola 2.10).

Ani jedna z aplikací neumožňuje editaci prostorových údajů uložených v souboru.

2.11.5 gvSIG Mobile

GvSig Mobile[24] je GIS zaměřený na mobilní zařízení. Aplikace podporuje velkou škálu vektorových i rastrových formátů, jejich editaci a následný export. Aplikace je dostupná pro: Windows Mobile 5.0 a 6.0.

Poslední verze aplikace 0.3 byla vydána 27.9.2010[25], lze tak předpokládat, že vývoj byl ukončen.

2.11.6 pcMapper

PcMapper[34] je jednoduchá aplikace pro editaci map. Aplikace umožňuje tvorbu bodů, čar, polygonů a přidávání textových poznámek. Výsledné mapy lze exportovat do KML[27]. Aplikace je dostupná pro operační systém Android a je nutné si pro používání aplikaci zakoupit. K aplikaci je k dispozici placené databázové rozšíření[33], které přidává podporu atributů uložených v SQLite databázi a podporu práce se shapefile soubory.

Aplikace v základu zaostává za mGISi v podpoře shapefile formátu. Po zakoupení databázového rozšíření ovšem mGIS v dostupné funkčnosti převyšuje.

2.12 Knihovny třetích stran

Následující sekce popisují knihovny třetích stran pro implementaci GIS nástroje.

2.12.1 SQLite

SQLite [39] je relační databázový systém napsaný v jazyce C. SQLite je knihovna, které se přilinkuje do aplikace a stává se tak její součástí. Na rozdíl od databázových systémů typu klient-server odpadá nutnost samostatného procesu a složité komunikace mezi procesy. Celá databáze (definice, tabulky i data) jsou uloženy v jednom souboru nezávislém na platformě. Databáze implementuje jednoduchý systém zámků, kdy při zápisu dochází k uzamčení celého souboru. Čtení může být prováděno paralelně z více vláken.

Knihovna SQLite je použita společně s rozšířením Spatialite.

Zdrojové kódy knihovny jsou dostupné pod licencí public domain.

2.12.2 Spatialite

SpatiaLite [37] je open source rozšíření pro knihovnu SQLite přidávající podporu prostorových dat. S databází se pracuje pomocí dotazů v jazyce SQL. Prostorová data jsou ukládána ve formě binárních dat do sloupce datového typu BLOB(Binary large object). Pro práci s prostorovými daty se v SQL dotazech používají SpatiaLite funkce, které transformují binární podobu objektu srozumitelné podoby. Příkladem funkce je `AsText` pro převod binárních data do formátu Well Known Text (WKT). Bod se souřadnicemi x=10 a y=20 je v formátu WKT reprezentován řetězcem: POINT(10 20). V tabulce na obrázku 2.2 jsou uvedeny podporované prostorové objekty.

Součástí SpatiaLite je rozšíření, které umožňuje exportovat a importovat do databáze data ve formátu shapefile. Export a import je omezen pouze na typy objektů podporované shapefile formátem (Tabulka 2.2).

SpatiaLite je dostupná pod MPL tri-license[31]

2.12.3 SQLite JDBC Driver

SQLite JDBC [40] Driver je knihovna umožňující přístup k databázi SQLite v jazyce Java. Knihovnu vytvořil Taro L. Saito. Knihovna implementuje standartní rozhraní JDBC pro práci s databází v jazyce Java a umožňuje tak využívat běžné prostředky jazyka pro práci s databází.

2.12.4 OSMDroid

OSMDroid [32] je open source knihovna implementující API pro přístup k datům z dlaždicových serverů, primárně z OpenStreetMap v Android aplikacích. OSMDroid je kompatibilní se starší, ale doposud funkční, verzí Google Maps Android API (kapitola 2.4.3) a umožňuje tak zobrazovat i data získaná z Google Maps. Pro jednoduchou integraci do mobilní aplikace poskytuje knihovna komponentu `mapView`. S komponentou se v rámci aplikace pracuje jako s ostatními komponentami uživatelského rozhraní. Knihovna umožňuje v offline režimu pracovat s mapovými podklady uloženými na disku.

Kapitola 3

Mapování

Tato diplomová práce se zabývá vytvořením GIS mobilního nástroje pro sběr dat a mapování oblastí. Následující kapitola popisuje průběh mapování z teoretického hlediska, i průběh skutečného mapování vybrané oblasti provedeného v rámci této práce.

3.1 Etapy mapování

Vytvoření mapy vybrané oblasti může v závislosti na velikosti oblasti a množství mapovaných objektů zabrat několik hodin, ale také i několik dnů. Jedná se tedy o rozsáhlou činnost, kterou můžeme rozdělit do třech hlavních na sebe navazujících etap. Přípravu a plánování (kapitola 3.1.1) provedené v pohodlí domova na základě dostupných informací, mapování (kapitola 3.1.2) probíhající přímo v terénu a zpracování údajů (kapitola 3.1.3) po návratu z terénu.

3.1.1 Příprava a plánování

Teoreticky lze tuto etapu při mapování vynechat, ale vzhledem k tomu, že mapování velké oblasti může velmi náročné, důkladná příprava a plánování celý proces značně zjednodušuje. Před samotnou výpravou do terénu je důležité si určit cíl mapování, jaké geografické údaje budeme zaznamenávat. To nám dovoluje si předem připravit v aplikaci jednotlivé mapové vrstvy s relevantními atributy. Nikdy si nemůžeme být jisti, že v oblasti, ve které se budeme pohybovat, bude k dispozici připojení k internetu. Je dobré si tedy dopředu nachystat mapové podklady pro zvolenou oblast a přilehlé okolí.

Na základě nashromážděných informací o dané oblasti a určeném cíli mapování je dobré naplánovat si přibližnou trasu terénem. Tím se snižuje riziko, že se budeme muset procházet stejný úsek opakovaně. Pokud se jedná o rozlehlou oblast, můžeme do plánování zahrnout také dostupné dopravní prostředky nebo hromadnou dopravu.

3.1.2 Mapování

Na základě předchozí přípravy procházíme po zvolené oblasti s geozápisníkem (kapitola 3.2) a do předem vytvořených mapových vrstev vkládáme naměřená geografická data. Když dorazíme na místo, jehož polohu chceme zaznamenat, získáme pomocí zabudovaného GPS modulu souřadnice. V aplikaci dojde k vytvoření bodu ve vybrané mapové vrstvě, s příslušnými geografickými souřadnicemi. Alternativní možností je zadání bodu pomocí uživatelského rozhraní mGISmobile. V případě vrstvy typu bod je objekt tvořen souřadnicemi

tohoto bodu. Ovšem v případě vrstev typu linie nebo polygon, jsou jednotlivé objekty tvořeny posloupností bodů.

Po dokončení geometrie může nastat situace, kdy je nutné již vytvořený objekt zpřesnit. V takovém případě je možné v existujích objektech upravovat polohu jednotlivých bodů, body přidávát nebo odebírat.

Každá z předem připravených vrstev definuje atributy jednotlivých objektů. U objektů s kompletní geometrií lze hodnoty jednotlivých atributů zadat.

3.1.3 Zpracování údajů

Po dokončení mapování je nutné získaná data zpracovat, vytřídit a přenést do systému, který bude se získaními daty dále pracovat.

Úprava dat zahrnuje editaci nebo smazání nepotřebných objektů. V mobilní aplikaci mGISmobile nelze jednou vytvořené objekty smazat. Pouze se nastavuje příznak o smazání objekt, a proto je nutné jejich odstranění provést dodatečně. Úpravu geometrie nebo atributů lze provádět přímo na mobilním zařízení, ovšem stolní počítač a dostupný software poskytuje více možností pro práci s objekty a vyšší komfort práce.

Data lze z mobilní aplikace získat několika způsoby. Prvním způsobem je použít aplikaci, která umí data získávat pomocí standardu WFS (kapitola 2.9), a data si stáhnout přímo z aplikace mGISserver. Druhým způsobem je získat data z aplikace ve formě souboru, a ten přenést do cílové zařízení a softwaru. Zde se nám nabízejí dva způsoby, jak získat data v podobě souboru. Data jsou ukládána do databáze Spatialite, která je uložena v jediném souboru v paměti mobilního zařízení a stolního pc. Pokud aplikace umí pracovat s tímto typem databáze, můžeme data číst přímo z tohoto souboru. Další možností je export zvolené vrstvy do formátu shapefile a jeho následovné zpracování.

3.2 Koncept geozápisníku

Geodet využívá geozápisník v průběhu mapování k zaznamenávaní naměřených údajů. Při použití klasického papírového zápisníku potřebuje geodet navíc i zařízení pro určení polohy. Papírový zápisník umožní geodetovi zapsat potřebné údaje avšak neposkytuje příliš velký komfort práce. Při zaznamenání bodu musí geodet zjistit jeho polohu a tu následně opsat do zápisníku. Protože data v zápisníku jsou pouhá čísla, není pro geodeta možné, aby si v průběhu mapování již zaznamenaná data prohlédl. Velká nevýhoda papírového zápisníku se projeví při zpracování výsledných dat. Pro zpracování v elektronickém systému musí být data do systému přenesena, což často znamená ruční přespsání všech údajů do systému.

Elektronický geozápisník oproti papírovému umožňuje některé zbytečné kroky při zaznamenávání či zpracování dat vynechat. Geodet má zároveň možnost s daty interaktivně pracovat již v průběhu mapování. Pro komplexní použitý by měl geodetický zápisník nabízet následující funkce:

- Zjištění polohy – zápisník by měl být schopen určit aktuální polohu a dovolit geodetovi zaznamenat bod s aktuální polohou.
- Zobrazení zaznamenaných geoobjektů – umožňuje geodetovi vidět zaznamenané body jako celek a odhalit případnou chybu v datech na místě.
- Zobrazení podkladové mapy – podkladová mapa usnadňuje geodetovi pohyb po oblasti a umožňuje mu promítnou zaznamená oblasti do mapy.

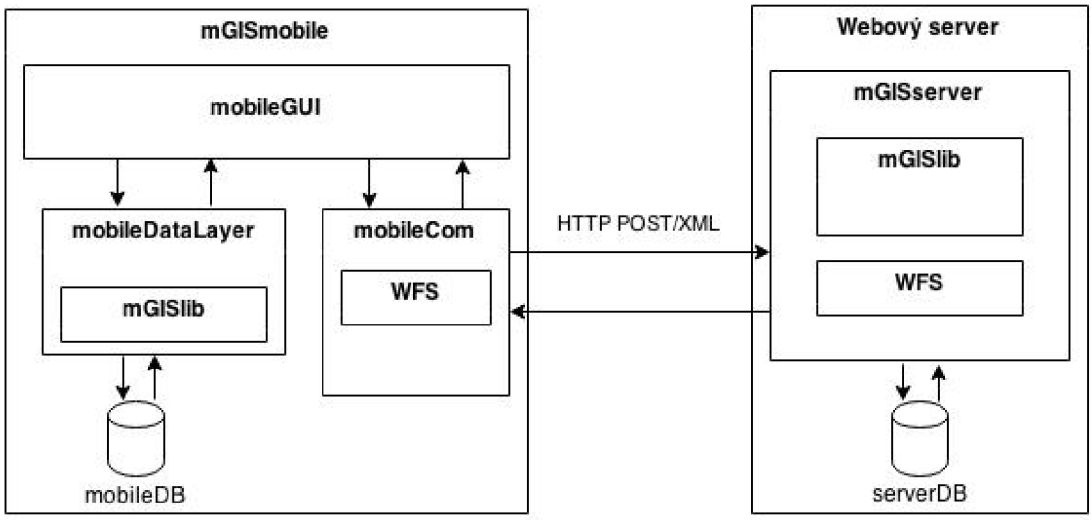
- Editaci geoobjektů – zápisník by měl geodetovi umožňit dodatečnou editaci geoobjektů, a to jak editaci geometrie, tak editaci atributů.
- Export dat – zápisník by měl disponovat možností pro přenost dat do systému, ve kterém bude probíhat jejich další zpracování.

Kapitola 4

Koncept mGIS

Tato kapitola rezebírá koncept mGIS z abstraktního hlediska a přibližuje průběh jednotlivých procesů v rámci aplikací. Následující kapitoly popisují:

- Architekturu mGIS – struktura aplikací.
- Datový model – popis struktury relační databáze.
- Elementární procesy – popis průběhu základních procesů při práci s vrstvami a geoobjekty.
- Komplexní procesy – paralelní mapování, synchronizace dat.
- Uživatelské rozhraní – přehled obrazovek aplikace mGISmobile.



Obrázek 4.1: Architektura mGIS.

4.1 Architektura

Na obrázku 4.1 vidíme nákres architektury aplikací mGISmobile a mGISserver. Aby bylo možné zajistit práci v offline režimu, je nutné, aby data byla perzistentně ukládána v obou

aplikacích. Jako perzistentní úložiště dat je v obou aplikacích použita relační databáze. Díky této skutečnosti obsahují obě aplikace totožné funkce pro práci s databází a používají stejnou vnitřní reprezentaci datových objektů. Společná funkcionalita je vyčleněna do knihovny mGISlib, kterou obě aplikace využívají.

4.1.1 mGISmobile

Aplikaci mGISmobile můžeme rozdělit do třech hlavních modulů:

- mobileGUI – uživatelské rozhraní aplikace je nejdůležitějším modulem. Kromě interakce s uživatelem modul zajišťuje propojení ostatních modulů a obsahuje mapovací logiku. Vzhled a funkce jednotlivých obrazovek uživatelského rozhraní je popsána v kapitole 4.5.

- mobileDataLayer – v datové vrstvě jsou po čas běhu aplikace uchovávány mapové vrstvy a geoobjekty načtené z databáze mobileDB. Data mapových vrstev a geoobjektů jsou uložena v datových objektech definovaných v knihovně mGISlib (kapitola 4.1.3). Modul poskytuje uživatelskému rozhraní prostředky pro práci s databází.

- mobileCom – komunikační modul poskytuje funkce pro komunikaci s aplikací mGISserver. Komunikace probíhá synchroně prostřednictvím protokolu HTTP. Jednotlivé zprávy jsou odesílány metodou POST. Součástí modulu jsou funkce pro převod mezi datovými objekty knihovny mGISlib a formátem XML.

4.1.2 mGISserver

mGISserver je implementován jako HTTP servlet běžící v rámci webového serveru, v tomto případě je použit Apache Tomcat [16]. mGISserver na rozdíl od mGISmobil neobsahuje žádné uživatelské rozhraní. Jedinou úlohou aplikace je zpracování WFS požadavků a práce s geoobjekty uloženými v databázi. Webový server zajišťuje navázání komunikace, příjem zpráv a odeslání odpovědi pomocí protokolu HTTP. Přijatá data jsou následně předána servletu ke zpracování. Po zpracování požadavku jsou data odpovědi předána webovému serveru, který zajistí jejich odeslání. Součástí aplikace jsou funkce pro převod mezi datovými objekty a formátem XML, podobné funkcím v mGISmobile. Tato funkcionalita nemůže být součástí mGISlib, protože dostupné prostředky pro zpracovaná XML se na platformách liší. Server pro reprezentaci geoobjektů a práci s databází využívá prostředky dostupné v mGISlib.

4.1.3 mGISlib

Knihovna mGISlib obsahuje objekty a funkce, které jsou společné pro mGISmobil i mGISserver. Vyčleněním objektů a funkcí do samostatné knihovny se zabránilo zbytečné redundaci kódu a došlo ke zjednodušení implementace a údržby. Knihovna obsahuje datové objekty pro reprezentaci geoobjektů a mapových vrstev. Součástí knihovny jsou funkce pro vytvoření SQL dotazů pro vkládání, úpravu a čtení datových objektů z databáze. Z důvodu odlišného přístupu k databázi Spatialite na různých platformách, popsaného v kapitole 5.2, nelze do knihovny zahrnout kompletní práci s databází.

Datový objekt mapové vrstvy

Datový objekt pro reprezentaci mapové vrstvy obsahuje následující položky:

- Název – používá se jako jednoznačný identifikátor vrstvy v rámci mGIS.
- Typ geometrie – vrstvy mohou být typu bod, linie nebo polygon.
- Seznam atributů – vrstva definuje jaké atributy budou mít všechny geoobjekty vrstvy. Tento senzam obsahuje dvojice, kde je uveden název atributu a typ jeho hodnoty. Hodnoty atributů mohou být celá čísla, desetinná čísla, nebo text. Tento seznam může být prázdný.
- Seznam geoobjektů – geoobjekty, se kterými aplikace pracuje = byly načteny z databáze.

Datový objekt geoobjektu

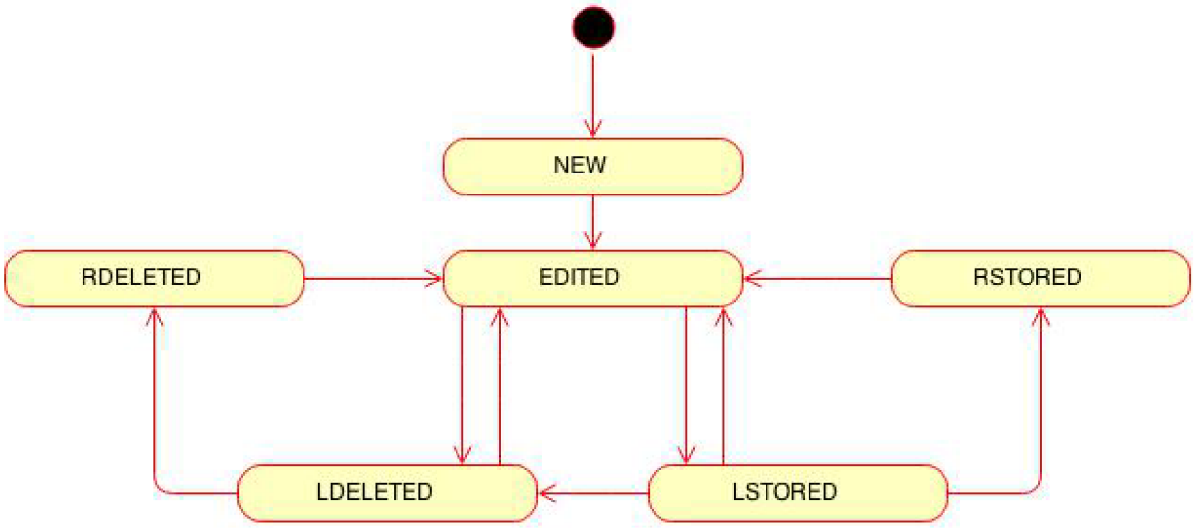
Datový objekt pro reprezentaci geoobjektu obsahuje následující položky:

- OID – jednoznačný identifikátor objektu v rámci mapové vrstvy.
- Typ geometrie – typ geometrie shodný s typem geometrie vrstvy.
- Geometrie – seznam jednotlivých bodů geometrie.
- Stav – určuje stav objektu v rámci aplikace. Výčet možný stavů je uveden níže.
- Seznam atributů – seznam obsahuje dvojici pro každý atribut definovaný v mapové vrstvě geoobjektu. Dvojice obsahuje název atributu a jeho hodnotu. Seznam je prázdný, pokud u mapové vrstvy žádné atributy neexistují.

V knihovně je definováno šest stavů, ve kterých se objekty mohou v průběhu života nacházet:

- NEW – nově vytvořený objekt v mGISmobil bez kompletní geometrie.
- EDITED – editovaný objekt nebo objekt, u kterého byla dokončena geometrie, ale ještě nebyl uložen do databáze
- LSTORED – objekt uložený v rámci databáze mGISmobil. Tento stav naznačuje, že objekt po vytvoření nebyl ještě odeslán na mGISserver, nebo došlo od jeho přijetí k úpravě. Při synchronizaci jsou tyto objekty odeslány na mGISserver.
- RSTORED – objekt uložený v databázi mGISserver, který nebyl změněn. Tyto objekty se při synchronizaci neodesílají.
- LDELETED – objekt, který byl uživatelem mGISmobile smazán, ale změna ještě nebyla odeslána na mGISserver. Při synchronizaci jsou tyto objekty odeslány na mGISserver.
- RDELETED – smazaný objekt uložený v databázi mGIServer, který nebyl obnoven. Tyto objekty se při synchronizaci neodesílají.

Na obrázku 4.2 je zobrazen diagram přechodů mezi jednotlivými stavy.



Obrázek 4.2: Diagram stavů objektu

4.2 Datový model databáze

Jako perzistentní datové úložiště slouží v aplikacích mGISmobile a mGISserver relační databáze SQLite s prostorovým rozšířením Spatialite. Základem relačních databází jsou relace (databázové tabulky), což jsou dvourozměrné struktury skládající se ze záhlaví a těla. Jednotlivé sloupce tabulky se nazývají atributy a mají určený konkrétní datový typ doménu, což je množina přípustných hodnot daného atributu. Řádky tabulky jsou poté záznamy. Řádky slouží k vlastnímu uložení dat.

Tabulky v databázích aplikací mGIS můžeme rozdělit do dvou skupin, podle toho, zda obsahují geoobjekty.

- Systémové tabulky – data v těchto tabulkách jsou nutná pro běh aplikace nebo ke zpracování mimo aplikace mGIS.
- Tabulky vrstev – tabulky obsahující geoobjekty vytvořené uživatelem.

4.2.1 Systémové tabulky

V databázi aplikací mGIS existují dvě systémové tabulky `GeomTables` a `CaptureTime`. Pokud tyto tabulky neexistují, jsou automaticky při startu aplikace vytvořeny.

Tabulka `GeomTables` slouží k uložení základních informací o dostupných mapových vrstvách. Záznamy v této tabulce se používají pro určení, jaké mapové vrstvy jsou uživateli k dipozici. Atributy tabulky jsou následující:

- ID – identifikátor záznamu.
- Název – jméno mapové vrstvy.
- Geometrie – typ geometrie vrstvy.

Tabulka `CaptureTime` slouží záznam průběhu mapování. Při přidání nového bodu do libovolného geoobjektu je do tabulky vložen záznam s aktuálním časem. Záznamy jsou do tabulky pouze přidávány a i když dojde k odstranění bodu, záznam není smazán. Data

z této tabulky nejsou v rámci aplikací mGIS využívána a slouží pro zpracování v externích programech. Data z této tabulky mohou být využita například pro analýzu průběhu mapování. Tabulka obsahuje následující atributy:

- ID – identifikátor záznamu.
- Lat – zeměpisná šířka bodu.
- Lon – zeměpisná délka bodu.
- Vrstva – jméno vrstvy, do které byl bod přidán.
- Čas – čas zaznamenání bodu.

4.2.2 Tabulky vrstev

V kapitole 2.4.1 bylo zmíněno, že typickým přístupem je uložení geometrie a atributů geoobjektu odděleně. V rámci aplikací mGIS je využit jiný přístup a geometrie a atributy geoobjektu jsou uloženy společně v jednom záznamu tabulky. Aplikace pracují s předpokladem, že jednotlivé vrstvy jsou homogení, a tedy všechny geoobjekty v dané vrstvě mají shodný typ geometrie i stejné atributy. V takovém případě můžeme každou mapovou vrstvu definovat jako tabulku relační databáze, ve které existuje jeden sloupec pro uložení geometrie a jeden sloupec pro každý požadovaný atribut goeobjektu. Aby bylo možné atributy od sebe rozlišit, jsou sloupce pojmenovány názvy jednotlivých atributů.

Struktura tabulky vrstvy s jedním atributem v aplikacích mGIS vypadá následovně:

- OID – identifikátor záznamu.
- Geometry – geometrie geoobjektu.
- Stav – stav geoobjektu.
- Atribut – hodnota atributu.

Uložení geometrie a atributů v rámci jedné tabulky má význam i z pohledu exportu dat. Použité prostorové rozšíření databáze umožňuje export tabulky do formátu shapefile pomocí vestavěné funkce. Protože všechna data jsou uložena v jedné tabulkce, výsledný soubor obsahuje jak geometrii objektů, tak jejich atributy. Pokud by byl použit konvenční přístup a data by byla uložena odděleně, bylo by možné za použití vestavěné funkce exportovat do formátu shapefile pouze geometrii geoobjektů.

4.3 Elementární procesy

Za elementární procesy považujeme procesy, které uživatel provádí běžně při práci s aplikací a týkají se jednoho geoobjektu nebo jedné mapové vrstvy. Z pohledu práce s mapovými vrstvami se jedná o jejich vytvoření a smazání, u geoobjektů o vytvoření a úpravu. Pod pojmem úprava geoobjektu rozumíme v rámci mGIS změnu geometrie, změnu atributů, smazání nebo obnovení smazaného geoobjektu.

Při práci s aplikací mGISmobile uživatel vždy může v jednu chvíli pracovat pouze s geoobjekty v rámci jedné vrstvy. Vrstvu, ve které uživatel pracuje, označujeme jako aktivní. Při

startu aplikace je jako aktivní vrstva označena první v pořadí. Pokud si uživatel přeje pracovat s objekty jiné vrstvy, změní aktivní vrstvy pomocí bočního panelu hlavní obrazovky 4.11.

Při spuštění aplikace dochází ke kontrole, zda existuje v mobilním zařízení soubor s databází. Pokud soubor neexistuje, dojde k jeho vytvoření. Dalším krokem je načtení dostupných mapových vrstev a jejich uložení v modulu `mobileDataLayer`. V tuto chvíli mapové vrstvy neobsahují žádné geoobjekty. Po získání aktuální polohy nebo při zobrazení nové oblasti dojde k načtení objektů z databáze. Načítají se pouze objekty, které jsou zobrazeny na obrazovce zařízení, a jsou uloženy do příslušné mapové vrstvy.

4.3.1 Vytvoření mapové vrstvy

Při vytváření nové vrstvy uživatel zadává v rámci uživateleského rozhraní jméno vrstvy, typ geometrie a případně atributy objektů. V modulu `mobileGUI` dojde k vytvoření objektu vrstvy, kterému je nastaven příslušný typ geometrie, jméno, a je naplněn seznam atributů. Výsledný objekt je uložen pod jménem vrstvy do modulu `mobileDataLayer`. Následně je pro novou vrstvu vytvořena tabulka v databázi (kapitola 4.2).

4.3.2 Smazání mapové vrstvy

Na základě požadavku uživatele je podle jména získán datový objekt vrstvy z modulu `mobileDataLayer`. Mapová vrstva je následně odebrána ze seznamu dostupných vrstev a vrstva i se všemi daty, která obsahuje, je smazána z databáze. Smazání vrstvy je jediný způsob, jak smazat data v aplikacích mGIS permanentně a tento proces je nevratný.

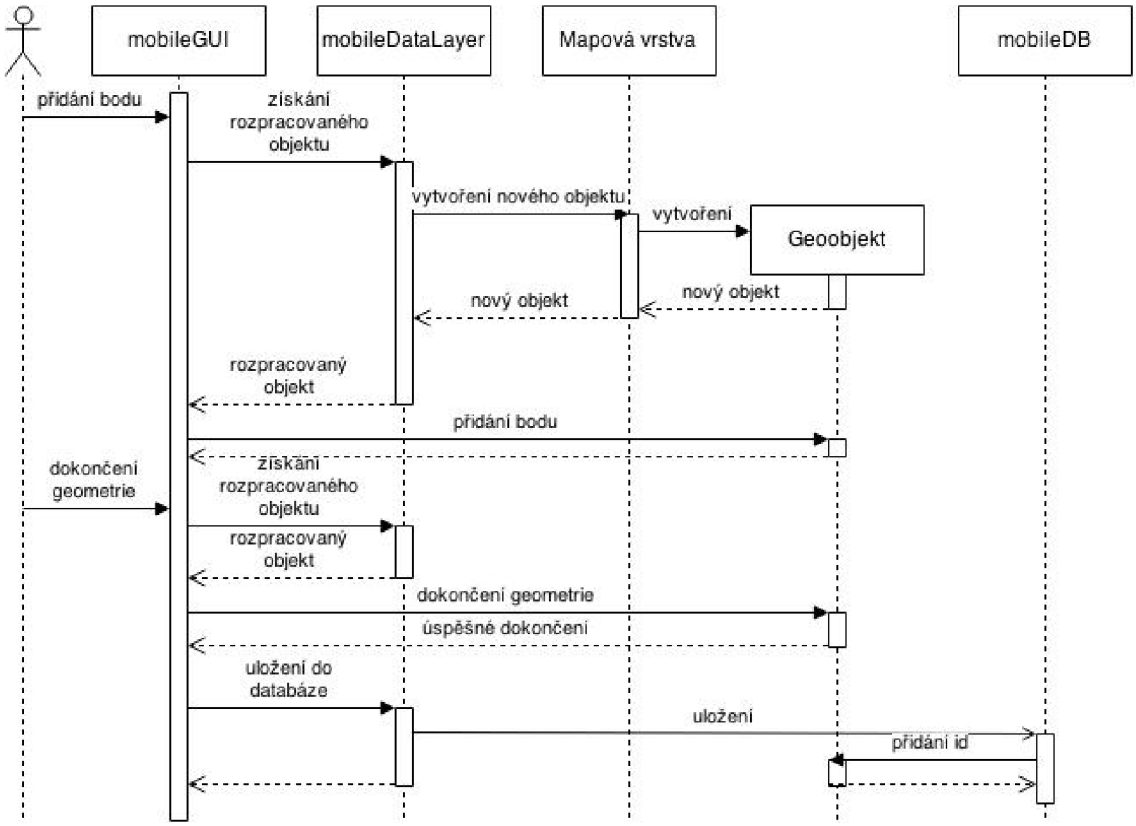
4.3.3 Vytvoření nového geoobjektu

Průběh vytvoření nového objektu je znázorněn sekvenčním diagramem na obrázku 4.3. Tvorba nových objektů probíhá vždy v aktivní mapové vrstvě. Při požadavku uživatele na uložení souřadnic bodu je z modulu `mobileDataLayer` získán rozpracovaný geoobjekt pro danou vrstvu. Rozpracovaným geoobjektem je nově vytvořený geoobjekt, u kterého stále probíhá zadávání bodů jeho geometrie. Pokud v aktivní vrstvě žádný neexistuje, je vytvořen. Nově vytvořený geoobjekt má typ geometrie shodný s typem vrstvy a stav objektu je nastaven na `NEW`. Ropracovaný objekt nemá OID, ten je mu přiřazen až při uložení do databáze. Při přidávání dalších bodů geometrie je z datové vrstvy získán již existující objekt, do kterého je bod přidán.

Na základě požadavku od uživatele na dokončení objektu dojde ke kontrole jeho geometrie. V případě linie musí geometrie mít alespoň dva body, u polygonu musí být body minimálně tři. Pokud geometrie splňuje podmínky pro dokončení objektu, je objekt v aktivní vrstvě dokončen. Dokončený objekt má kompletní geometrii a může být uložen. V tuto chvíli je objekt odebrán z rozpracovaných objektů v datové vrstvě a je umístěn mezi objekty příslušné vrstvy. Ve stejnou chvíli dojde i k uložení objektu do databáze. Při uložení objektu je vygenerován celočíselný identifikátor objektu, který do objektu uložen.

4.3.4 Úprav geoobjektu

Proces úpravy objektu je znázorněn sekvenčním diagramem na obrázku 4.4. Úprava, mazání a obnova geoobjektů probíhá obdobně jako u vytváření objektu v aktivní vrstvě. Uživatel



Obrázek 4.3: Sekvenční diagram vytvoření objektu.

v první řadě vybere objekt, se kterým bude pracovat. Při kliknutí uživatele na display mobilního zařízení dojde v rámci modulu `mobileDataLayer` k výběru příslušného geoobjektu. Pokud se v aktviní vrstvě v místě kliknutí nachází nějaký geoobjekt, je vybrán. Příslušný geoobjekt je poté v modulu `mobileDataLayer` označen jako vybraný.

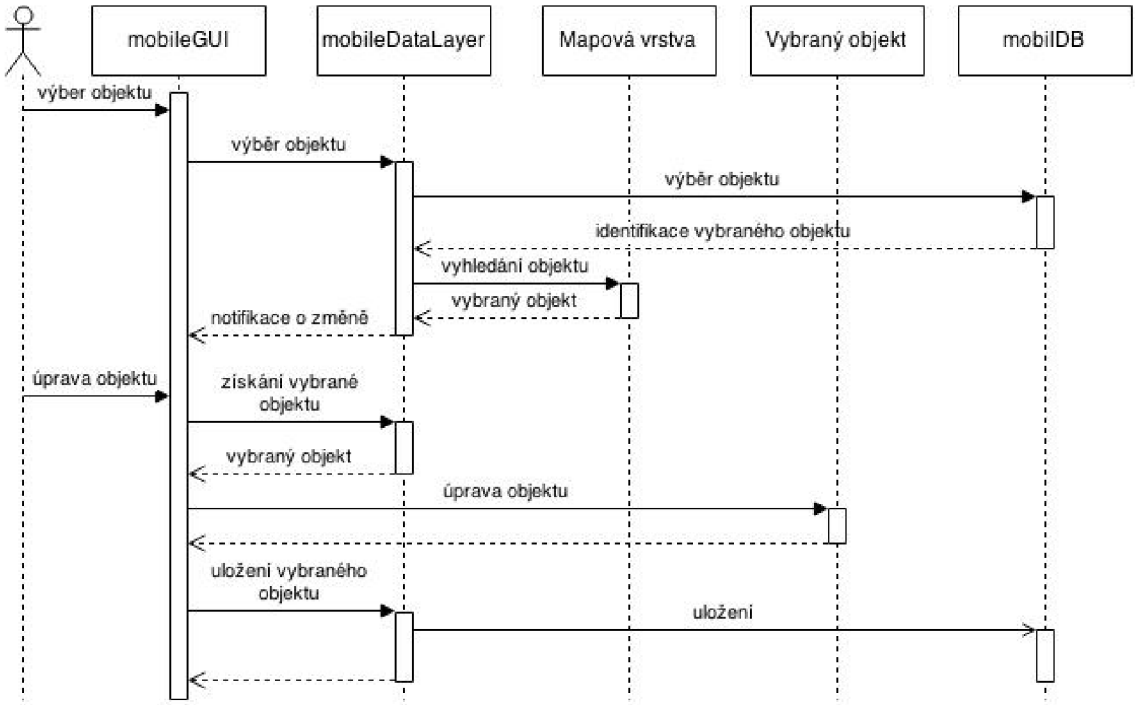
Uživatel prostřednictvím uživatelského rozhraní provede požadovanou změnu. Tato změna se promítne v datech daného geoobjektu a změny jsou následně prostřednictvím modulu `mobileDataLayer` uloženy do databáze. Smazáním nebo obnovením se rozumní nastavení příslušného stavu objektu, z pohledu zpracování se tedy jedná o stejný proces, jako je úprava objektu.

4.4 Komplexní procesy

Jako komplexní bereme v rámci souboru mGIS procesy, které pracují zároveň s několika geoobjekty. Mezi tyto procesy můžeme zařadit funkci paralelního mapování, která nám dovoluje vytvářet několik objektů současně, a synchronizaci dat mezi aplikacem mGISmobile a mGISserver. Popis a průběh těchto procesů je popsán v následujících sekcích.

4.4.1 Paralelní mapování

Při mapování můžeme postupovat sekvenčně tak, že si zvolíme objekt a následně zmapujeme jeho kompletní geometrii. Až po dokončení vybraného objektu se posuneme k mapování dalšího objektu. Tento postup je zcela korektní, ovšem v žádném případě není optimální.



Obrázek 4.4: Sekvenční diagram úpravy objektu.

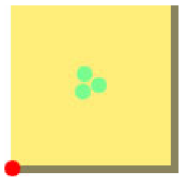
Nabízí se možnost mapovat objekty paralelně. Když zaměříme bod patřící současně do několika objektů, pak ho zaneseme do všech příslušných objektů. Tímto postupem se nám zkrátí čas potřebný pro mapování i vzdálenost, kterou musíme urazit. Pokud budeme uvažovat mapování rozlehlé oblasti, může být čas potřebný pro sekvenční mapování několikanásobně větší než při použití paralelního mapování.

Aby bylo možné v aplikaci mGISmobile realizovat paralelní mapování, jsou rozpracované objekty uloženy odděleně od ostatních objektů vrstvy. Jak bylo posáno v kapitlole 4.3.3, rozpracované objekty jsou uloženy v datové vrstvě a identifikovány jménem vrstvy, do které náleží. Při ukládání souřadnic nového bodu je z datové vrstvy získán rozpracovaný objekt v právě aktivní vrstvě. Poté je bod přidán do získaného objektu. Tento přístup nám dovoluje libovoně mezi vrstvami přepínat a přidávat body do několika objektů současně, ale limituje nás na jeden rozpracovaný objektu pro jednu vrstvu.

Srovnání sekvenčního a paralelního mapování

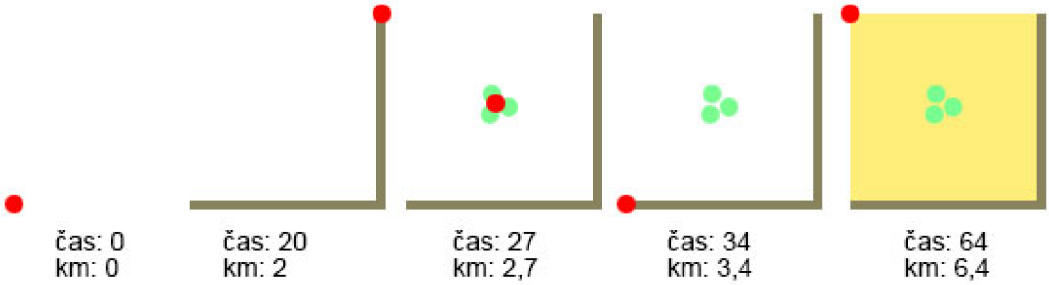
Srovnávání sekvenčního a paralelního mapování provedeme na ukázkovém příkladu zobrazeného na obrázku 4.5. Jako příklad mějme oblast obsahující louku se třemi stromy umístěnými uprostřed louky a cestu vedoucí okolo jižní a východní strany louky. Pro jednoduchost uvažujeme čtvercovou louku o délce hrany 1 km a geodet (člověk provádějící mapování) se pohybuje rychlostí 6 km/h. Geodet začíná mapovat v jihozápadním rohu louky, jeho pozice je označena červenou značkou. Čas potřebný pro zaznamenání bodu zanedbáme.

Na obrázku 4.6 vidíme průběh tvorby mapy dané oblasti, kdy geodet nejprve zaznamená geometrii cesty, poté stromy uprostřed louky a nakonec geometrii celé louky. Geodet začíná mapování v čase 0 zanesením prvního bodu cesty. Geodetovi následně zabere 20 minut, než zmapuje geometrii cesty. Poté se vydá nejkratší cestou směrem ke stromům, kam dorazí po 7 minutách a po dalších 7 minutách se ocitne v bodě, ve kterém začínal. V tuto chvíli mu



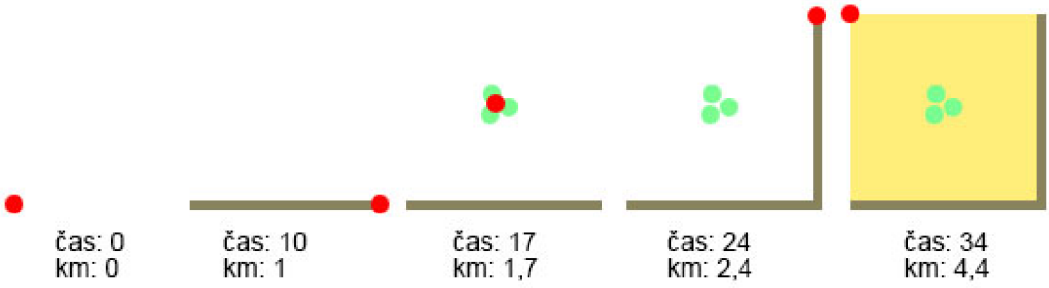
Obrázek 4.5: Ukázka mapované oblasti.

zbývá zmapovat geometrii louky, což mu zabere celkem 30 minut. Na posledním obrázku vidíme, že geodet skončil v severozápadním rohu louky, na zmapování potřeboval 64 minut a ušel 6,4 km.



Obrázek 4.6: Průběh vzniku mapy sekvenčním přístupem.

Na obrázku 4.7 vidíme průběh paralelního mapování stejné oblasti. Geodet v čase 0 zaznamená první souřadnici louky i cesty. Po 10 minutách se nachází v jihovýchodním rohu mapy, kde opět zaznamená souřadnice rohu louky a cesty. V tuto chvíli geodet vyrazí do středu louky, kam dorazí za 7 minut. Následně se vydá do severovýchodního rohu louky, kde zaznamená poslední bod cesty, roh louky. Po dalších 10 minutách geodet dorazí do severozápadního rohu louky a ukončuje mapování. Na rozdíl od sekvenčního přístupu zabralo geodetovi mapovaní pouhých 34 minut a ušel 3,4 km. Na tomto příkadu vidíme, že čas i vzdálenost při použití sekvenčního mapování je téměř dvojnásobná.



Obrázek 4.7: Průběh vzniku mapy paralelním přístupem.

Automatické zaznamenávání bodů

Aplikace mGISmobile nabízí pro účely paralelního mapování automatické zaznamenávání bodů. Automatické zaznamenávání je možné využít ve vrstvách typu linie a polygon. U vrstev typu bod nemá tato funkcionalita význam, protože objekt v této vrstvě sestává pouze

z jednoho jediného bodu. Docházelo by tedy k vytváření velkého množství nových objektů.

Automatické zaznamenávání bodů je možné zapnout tlačítkem u příslušné vrstvy v postranním panelu uživatelského rozhraní (kapitola 4.5.3). Pokud ve vrstvě neexistuje rozpracovaný objekt, je vytvořen, a jako první bod je přidána aktuální poloha. Od této chvíle jsou po nastavených časových intervalech do rozpracované geometrie přidávány body se souřadnicemi získanými z GPS. Tuto funkci je možné zapnout u několika vrstev najednou a vytvářet tak objekty, které mají společné body.

Uživatel může tuto funkci funkci kdykoli vypnout pomocí tlačítka v postranním panelu. Pokud uživatel dokončí rozpracovaný objekt, nebo mobilní zařízení ztratí signál se satelity GPS, je tato funkce vypnuta automaticky.

4.4.2 Synchronizace dat

Synchronizaci data mezi aplikacemi mGIS zajišťuje aplikace mGISmobile pomocí zpráv definovaným standartem WFS (kapitola 2.9). Průběh celého procesu je naznačen na obrázku **??**. V rámci aplikace mGISmobile je naznačena pouze spolupráce modulů `mobileDataLayer` a `mobileCom`, které mají na starost poskytnutí dat a komunikaci. Z pohledu aplikace mGISmobile můžeme sychnronizaci rozdělit na následující části:

- Připojení a ověření serveru – při vstupu na obrazovku pro synchronizaci dat (kapitola 4.8), se aplikace mGISmobile připojí k server mGISserver a pošle dotaz na operaci `GetCapabilites`. V tuto chvíli dojde k ověření správnosti IP adresy serveru a z dat odpovědi je ověřeno, že se jedná o správný server. Pokud je vše v pořádku, jsou uživateli zobrazeny mapové vrstvy dostupné pro synchronizaci.
- Odeslání dat – v tomto kroku jsou ve zvolených vrstvách odeslány všechny nově vytvořené nebo upravené geoobjekty. Aplikace tyto geoobjekty vybere na základě jejich stavu (kapitola 4.2). Odeslány jsou všechny geoobjekty, jejichž stav je nastaven na `LSTORED` nebo `LDELETED`.
- Stažení aktuálních dat – posledním krokem je stažení aktuálních dat ze serveru. Protože v rámci mGISserver nejsou implementovány omezují podmínky dotazů WFS, jsou staženy vždy všechna data synchronizované vrstvy.

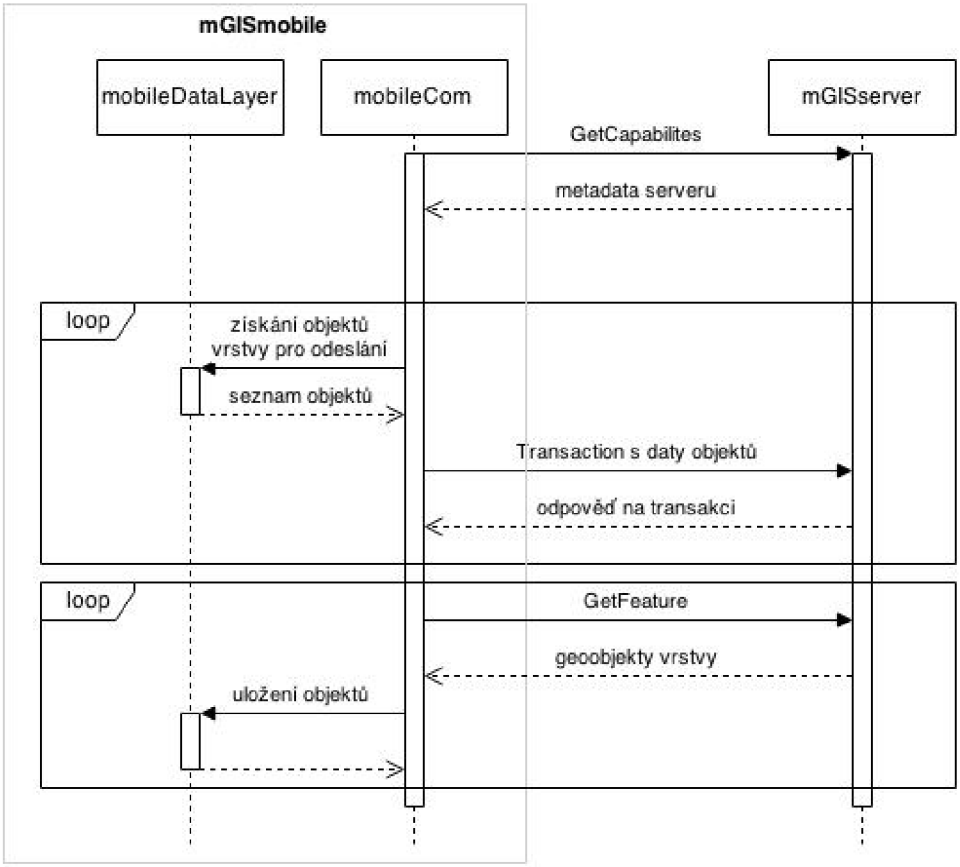
4.5 Uživatelského rozhraní mGISmobile

4.5.1 Mobilní vs. desktopové uživatelské rozhraní

Mobilní a desktopové aplikace se mezi sebou v mnohém odlišují. Každá platforma má své specifické ovládání, velikost obrazovky, dostupný hardware a mnohé další [30]. V následujích sekcích jsou popsány nejdůležitější odlišnosti, které je třeba při návrhu mobilního uživatelského rozhraní brát v potaz.

Ovládání myší vs. prsty

Pravděpodobně největším rozdílem, který můžeme mezi mobilními a PC aplikacemi nalézt, je způsob ovládání. Zatímco PC aplikace se zpravidla ovládají pomocí kurzoru myši, u mobilních aplikací je standardem ovládání pomocí doteků. Na obrázku 4.9 vidíme srovnání plochy prstu při doteku obrazovky a plochy kurzoru sloužící k výběru ovládacího prvku.



Obrázek 4.8: Průběh synchronizace dat.

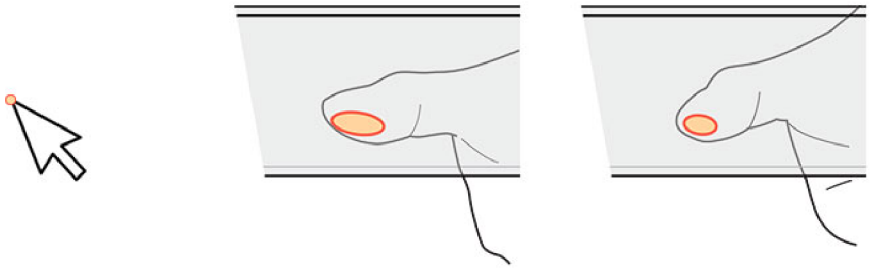
Jak vidíme, plocha při doteku je několikanásobně větší než u kurzoru a její velikost není konstantní. Velikost plochy závisí na úhlu prstu a na velikosti prstu daného jedince. Ovládání mobilních aplikací musí být těmto skutečnostem přizpůsobeno. Zatímco u PC aplikace si může dovolit malé ovládací prvky naskládané v těsné blízkosti, mobilní aplikace by se stala nepoužitelnou. Ovládací prvky na mobilním zařízení musí být dostatečně velké pro pohodlné ovládání, a aby se zamezilo nechtěnému výběru jiného prvku.

Dalším rozdílem je způsob, jakým můžeme s aplikací interagovat. U myši je samozřejmostí přítomnost pravého a levého tlačítka, které jsme schopni od sebe rozlišit. To nám dává možnost nabídnou uživateli dvě různé funkce. V případě prstu však nemáme možnost rozlišit o jaký prst se jedná, ovšem máme k dispozici prostředky pro sledování pohybu prstu po obrazovce. Díky tomu vznikla ustálená gesta [8] pro ovládání mobilních aplikací, příkladem může být přiblížení nebo oddálení pohledu pohybem prstů po obrazovce.

Klávesnice vs virtuální klávesnice

Hardwarová klávesnice u desktopového počítače poskytuje pohodlný způsob pro rychlé zadávání dat do aplikace. U drtivé většiny desktopových aplikací je samozřejmostí využití klávesových zkratek, pro zrychlení a zjednodušení práce s aplikací.

Současná mobilní zařízení nejsou hardwarovou klávesnicí vybavena a pro zadávání vstupu se používá klávesnice virtuální. Virtuální klávesnice zabírá značnou část dipleje a omezuje tak plochu pro zobrazení aplikace při zadávání vstupu. Při psaní za pomoci virtuální klá-



Obrázek 4.9: Srovnání velikosti výběrové plochy kurzoru a prstu.

vesnice často vznikají chyby. Z těchto důvodů je doporučeno se vyhnout použití virtuální klávesnice, pokud je to možné [30]. Místo zadání hodnoty může aplikace uživateli nabídnou například výběr z připraveného seznamu.

Velikost displeje

V porovnání s desktopovou aplikací běží mobilní aplikace na mnohem menší obrazovce. Okno aplikace je tudíž menší, ale ovládací prvky musí být dostatečně velké pro pohodlné ovládání prsty. Z toho vyplývá, že aplikace by měla uživateli zobrazovat pouze důležité funkce a ostatní funkcionalita by měla být skryta. Ostatní funkcionalita může být umístěna do menu nebo vyčleněna do samostatných obrazovek. V průběhu návrhu aplikace musí návrhář myslet na strukturu jednotlivých obrazovek a navigaci mezi nimi.

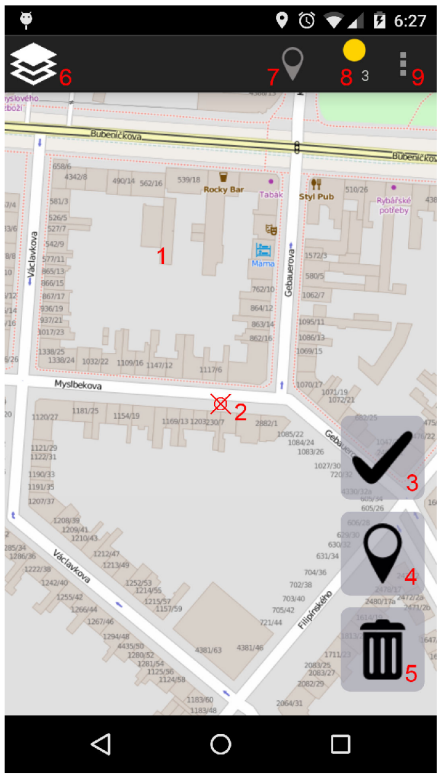
Změna orientace obrazovky

Uživatel má možnost mobilní zařízení používat ve vertikální nebo horizontální poloze. V systému Android je vertikální poloha označována jako portrait a horizontální jako landscape. Uživatel předpokládá, že aplikace mu umožní pracovat jak v režimu portrait, tak v režimu landscape. Při návrhu uživatelského rozhraní by měly být oba režimy brány v potaz.

4.5.2 Hlavní obrazovka

Na obrázku 4.10 je zobrazeno rozložení hlavního okna aplikace. Hlavním prvkem okna je plátno s vykreslenou podkladovou mapou a vytvořenými objekty, označené číslem 1. Aktuální pozice zařízení je označena červeným křížkem na mapě (číslo 2). K pohybu po mapě slouží běžně používaná gesta. Pomocí gesta Drag [8] se uživatel pohybuje po mapě, gesto Pinch [8] slouží k přiblížení a oddálení pohledu. Kliknutím uživatel může v právě aktivní vrstvě vybrat existující objekt pro jeho editaci. U vybraného objektu může uživatel vybrat libovolný z bodů a pomocí gesta Drag změnit jeho polohu.

V pravém dolním rohu okna se nachází trojice tlačítek pro práci s objekty. Tlačítko 3 slouží k dokončení geometrie rozpracovaného objektu v aktivní vrstvě. Tlačítko 4 slouží k přidání bodu do geometrie objektu. Při stisku tlačítka je přidán bod s aktuální polohou získanou pomocí GPS, při dlouhém stisknutí zadá uživatel polohu kliknutím na mapu. Pokud není vybrán žádný objekt, je nový přidán do geometrie ropracovaného objektu. V opačném případě je přidán do geometrie vybraného objektu. Tlačítko 5 slouží k odebrání bodu nebo celého objektu. Při stisknutí je odebrán poslení bod z geometrie rozpracovaného objektu nebo bod u vybraného objektu. Dlouhý stisk slouží ke smazání celého objektu.



Obrázek 4.10: Hlavní okno aplikace mGISmobile.

V horní části obrazovky se nachází informační panel. Tlačítko 6 na levém okraji slouží k otevření postraního panelu. Postranní panel je možné gestem Drag, které začne u levého okraje obrazovky. Při stiknutí tlačítka 7 je mapa vycentrována na aktuální pozici. Při podržení tlačítka 6 dojde k aktivaci sledovacího režimu. V tomto řežimu aplikace udržuje aktuální pozici zařízení ve středu obrazovky. Číslem 8 je označena indikace stavu GPS. Číslo u indikátoru informuje uživatele a počtu satelitů, které se podílejí na získávání polohy. Tlačítko 9 následně otevírá menu s dodatečným nastavením aplikace.

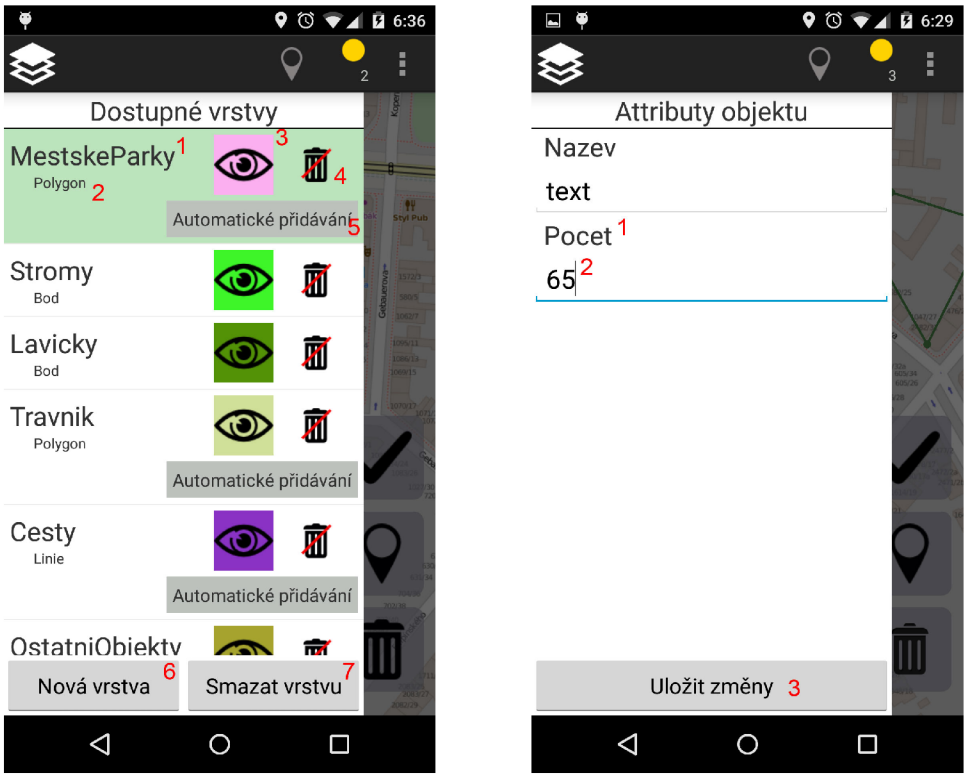
4.5.3 Postranní panel hlavního okna

Boční panel slouží k práci s vrstvami a atributy objektů. Pokud není vybrán žádný objekt, boční panel po otevření obsahuje seznam dostupných vrstev (obrázek 4.11 – vlevo). V opačném případě obsahuje panel seznam atributů a jejich hodnot (obrázek 4.11 – vpravo).

Senam vrstvev

V panelu je zobrazen seznam dostupných mapových vrstvev. U každé vrstvy je v levé polovině řádku zobrazeno jméno (číslo 1) a typ geometrie vrstvy (číslo 2). V pravé části řádku se nachází tlačítko 3 pro přepnutí viditelnosti objektů vrstvy. Tlačítko 4 slouží k přepínání viditelnosti smazaných objektů. U vrstev typu linie a polygon je dostupné tlačítko 5, které slouží k zapnutí a vypnutí funkce automatického přidávání bodů (kapitola 4.4.1).

Spodní část panelu obsahuje tlačítka pro vytvoření nové vrstvy (číslo 6) a smazání vrstvy (číslo 7).



Obrázek 4.11: Boční panel aplikace mGISmobile.

Seznam atributů

Panel atributů obsahuje výpis jmen (číslo 1) a hodnot jednotlivých atributů (číslo 2). Pokud uživatel změní hodnotu některého z atributů, je nutné změny uložit tlačítkem 3.

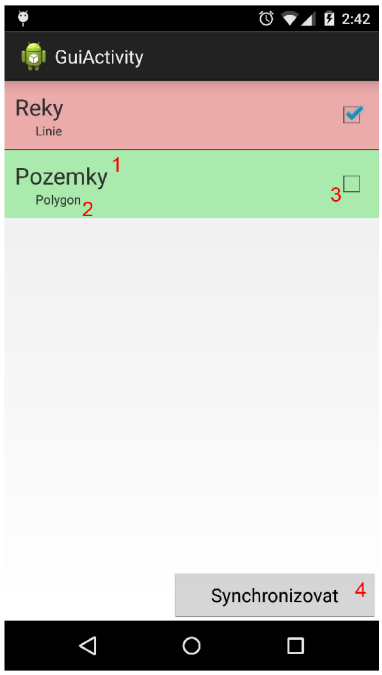
4.5.4 Obrazovka vytvoření vyrstvy

Obrázek 4.12 zobrazuje okno pro vytvoření nové mapové vrstvy. Povinnými údaji pro vytvoření nové vrstvy jsou její název a typ geometrie objektů. Typ uživatel vybírá z předem definovaných možností. Tlačítkem 1 uživatel může k nové vrstvě přiřadit atribut. U atributu je nutné zadat název a typ hodnoty, který bude možné zadat. Uživatel vybere jeden z předem definovaných typů. Stiskem tlačítka 2 dojde k vytvoření nové mapové vrstvy. Tlačítko 3 slouží k návratu na hlavní obrazovaku bez vytvoření vrstvy.

4.5.5 Obrazovka synchronizace s databází

Aplikace poskytuje uživateli jednoduché rozhraní pro synchronizaci objektů v mobilním zařízením s objekty na vzdáleném serveru. Na obrazovce, kterou vidíme na obrázu 4.13, je uživateli zobrazen seznam dostupných pamových vrstev. U každého záznamu je na lévé straně uvedeno jméno vrstvy (číslo 1) a typ (číslo 2). V pravé části seznamu checkbox pro výběr vrstvy (číslo 3). Jednotlivé vrstvy v seznamu jsou barevně odlišeny. V zelené vrstvě nedošlo k žádným změnám od stažení ze serveru. Žluté vrstvy obsahují nové nebo editované objekty a červené vrstvy se nenachází v mobilním zařízení. Tlačítkem v pravém spodním rohu obrazovky (číslo 4) dojde k synchronizaci u vybraných vrstev.

Obrázek 4.12: Obrazovka aplikace mGISmobile pro vytvoření nové vrstvy.



Obrázek 4.13: Obrazovka aplikace mGISmobile pro synchronizaci dat.

Kapitola 5

Implementace

5.1 Implementace architektury

V této kapitole jsou popsány implementační detaily architektury mGIS pospané v kapitole 4.1.

5.1.1 Implementace mGISmobile

Mobilní aplikace mGISmobile je implementována v jazyce Java a je určena pro operační systém Android. Modul `mobileGUI` je implementován v rámci tří *Aktivit* [2] systému Android. *Aktivita* je základní stavební prvek Android aplikací, který poskytujuje uživateli okno pro interakci a zajišťuje nějakou činnost v rámci aplikace. V mGISmobile existují následující *aktivity* a jejich funkce:

- GuiActivity – hlavní okno obrazovky. V rámci této aktivity probíhá veškerá manipulace s geometrií a atributy geoobjektů.
- NewLayerActivity – vytvoření nové vrstvy.
- SynchronizeActivity – synchronizace dat vrstev. V rámci této aktivity probíhá komunikace s mGISserver.

Jak bylo řečeno v kapitole 4.1, všechny geoobjekty a vrstvy jsou v aplikaci uloženy v modulu `mobileDataLayer`. Tento modul je implementován pomocí návrhového vzoru Singleton. Díky tomu jsou data jednoduše dostupná pro všechny aktivity a jednotlivé ovládací prvky.

5.1.2 Implementace servletu mGISserver

Hlavní třídou aplikace mGISserver je třída MGISServer, která dědí od třídy HttpServlet. Třída HttpServlet zajišťuje základní zpracování dotazu na server. Při příchodu nového požadavku tato třída přečte hlavičku HTTP požadavku a podle metody požadavku zavolá příslušnou metodu. Pro obsluhu požadavků zaslaných metodou POST tak stačí ve třídě MGISServer implementovat pouze metodu `doPost()`. Ukázku implementace HTTP servletu vidíme na obrázku 5.1.

```
public class MGISServer extends HttpServlet {

  protected void doPost( HttpServletRequest request,
                         HttpServletResponse response)
        throws ServletException, IOException {

    //získání dat požadavku
    InputStream data = request.getInputStream();

    //zpracování požadavku
    try {
      String data = server.processRequest(data);

      //odeslání odpovědi
        response.getWriter().write(data);
    } catch (ParseException e) {
      //nastala chyba pri zpracovani odpovedi
      //zadna nebude odeslana
    }
   }
}
```

Obrázek 5.1: Implementace mGISserver pomocí HTTP servletu.

5.1.3 Implementace mGISlib

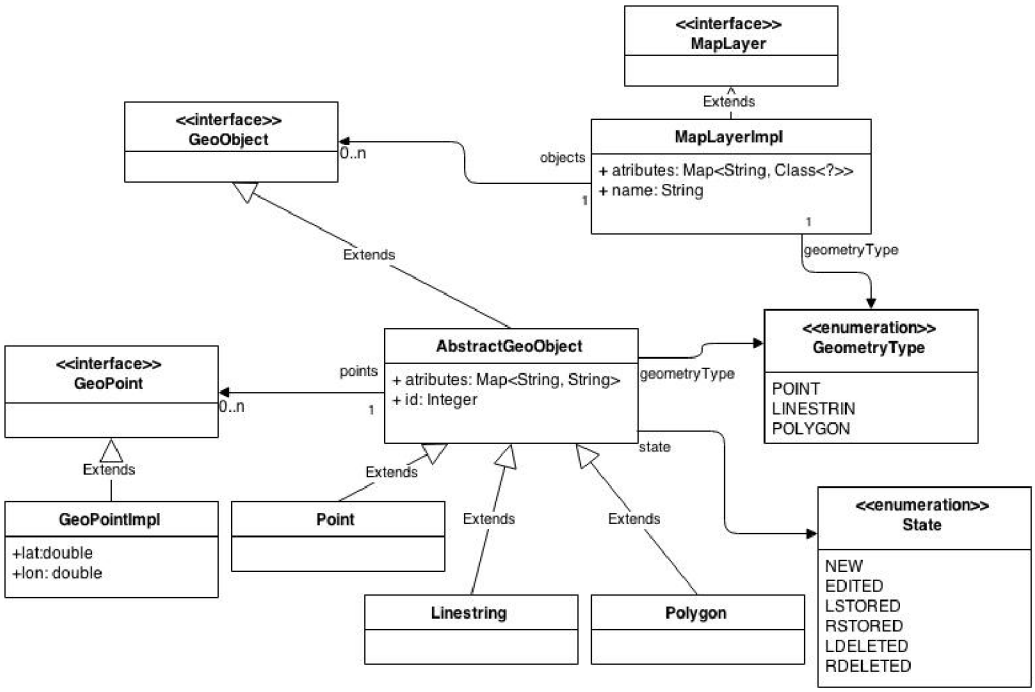
Z implementačního hlediska je mGISlib pouze soubor tříd. V rámci mGISlib jsou definovány datové objekty pro vnitřní reprezentaci mapových vrstev a geoobjektů. Struktura a vazby mezi jednotlivými objekty jsou znázorněny na obrázku 5.2.

Kromě datových objektů obsahuje mGISlib třídy `SQLQueriesImpl` a `DatatypesMaping`. Třída `SQLQueriesImpl` pro vytvoření SQL dotazu pro všechny operace prováděné nad databází v rámci souboru mGIS. Na základě požadované operace a datových objektů třída generuje řetězec SQL dotazu obsahujícího parametry. Z důvodů posaných v kapilole 5.2, nelze do knihovny zakomponovat tvorbu celého dotazu a proto doplnění hodnot pro jednotlivé parametry je zcela v režii příslušné aplikace.

Na obrázku 5.3 vidíme implementaci metod třídy `DatatypeMapping`. V rámci datových objektů jsou všechny hodnoty atributů uloženy v textové podobě a informace o jejich typu je uložena v objektu mapové vrstvy. Ovšem při vytváření tabulky vrstvy, jsou sloupce atributů vytvořeny se správným datovým typem. Tato třída poskytuje metody pro převod mezi databázovými typy a Java třídami jednotilvých typů.

5.2 Práce s databází

Aplikace mGISmobile i mGISserver používájí databázi SQLite s rozšířením Spatialite (kapitola 2.12.2) k perzistentnímu ukládání dat. Ovšem každá z těchto aplikací je určena pro jiný operační systém. Na různých operačních systémech se liší způsob jakým se aplikace připojuje k databázi a liší se třídy objektů používané při vykonávání dotazů. Následující kapitoly ukazují, jak probíhá práce s databází na operačních systémech Android a Windows.



Obrázek 5.2: Třídy definované v mGISlib a jejich závislosti.

```
public static String getDBType(Class<?> clazz) {
    if (clazz.equals(Double.class)) {
      return "REAL";
    }
    else if (clazz.equals(Integer.class)) {
      return "INTEGER";
    }
    else {
      return "TEXT";
    }
  }

  public static Class<?> getClassType(String type) {
    if (type.equals("INTEGER") || type.equals("INT")) {
      return Integer.class;
    }
    else if (type.equals("REAL")) {
      return Double.class;
    }
    else {
      return String.class;
    }
  }
```

Obrázek 5.3: Třída pro mapování Java tříd na datové typy databáze.

5.2.1 Práce s databází na sytému Android

Pro použití na operačním systému Android, je na stránkách projektu Spatialite-Android [38] dostupný port knihovny Spatialite a API třídy pro přístup k databázi. Pro jednoduchou integraci knihovny je na stránkách projektu je k dispozici ukázková aplikace s knihovnou v podobě přeložených binárních souborů. V ukázkové aplikaci je komentovaný zdrojový kód pro získání přístupu k databázi a provádění základních dotazů. Pro zkušené programátory jsou k dispozici zdrojové soubory knihovny pro překlad za pomoci nástroje Android NDK [15]. Na obrázku 5.4 je ukázka připojení k databázi a následného provedení dotazu.

```
Database database = new jsqlite.Database();
//otevření souboru s databází
database.open(spatialDbFile.getAbsolutePath(),
  jsqlite.Constants.SQLITE_OPEN_READWRITE |
  jsqlite.Constants.SQLITE_OPEN_CREATE);

try {
Stmt s= database.prepare(``SELECT ulice FROM domy WHERE barva = ?'');
  s.bind(1, ``modra'');
  while (s.step()) {
    //tisk ulice vybraného domu
    System.out.println(s.column_string(0));
  }
  s.close();
} catch (JSQLiteException e) {
  //chyba při provádění dotazu
}
```

Obrázek 5.4: Práce s databází na systému Android

Připojení k databázi

Připojení k databázi je velmi jednoduché. Připojení je udržováno v objektu třídy `Database`. Pomocí metody `open(file, mode)` je vytvořeno připojení k databázi. Prvním parametrem je cesta a názvem souboru, druhým parametrem je mód, ve kterém bude databáze otevřena. V ukázce je mód nastaven tak, že pokud databáze neexistuje, je vytvořena. V opačném případě je otevřena v režimu pro čtení a zápis.

Provádění dotazů

Pro provádění dotazů a parsování výsledků je určena třída `Stmt`. SQL dotazy mohou obashovat parametry, v tom případě je hodnota nahrazena otazníkem. Tento přístup nám umožňuje si dopředu připravit dotazy pro různé funkce aplikace a až při zpracování dotazu doplnit příslušné hodnoty. Pro nastavení hodnot jednotlivých parametru slouží metoda `bind(pos, value)`. Prvním parametrem je pozice parametru, druhým hodnota, která bude nahrazena místo otazníku. Pozice parametrů jsou číslovány od 1, ne od 0, jak je tomu v programovacích jazycích zvykem. Po nastavení hodnot všech parametrů je dotaz připraven ke zpracování. Pokud při zpracování dotazu dojde k chybě, je vyhozena výjimka. Všechny operace proto musí být prováděny v rámci `try-catch` bloku.

Voláním metody `step()` je proveden jeden krok zpracování dotazu. Pokud bylo provedení úspěšné, metoda vrací hodnotu `TRUE`, v opačném případě `FALSE`. Při každém volání metody zpracován jeden záznam. K následnému přečtení hodnot záznamu slouží metody s prefixem `column_`. Pro každý datový typ existuje příslušná metoda, například metoda `column_int(col)` pro získání hodnoty typu `int`. Parametrem metody je pořadí sloupce. Na rozdíl od metody `bind(pos, value)`, kde je pořadí číslováno od 1, zde je první sloupec na pozici 0.

Po zpracování všech hodnot by měl programátor objekt `Stmt` uzavřít metodou `close()`. Uzavřením objektu jsou uvolněny zdroje využité v rámci databáze. Protože objekt neimplementuje rozhraní `AutoCloseable` [9], musí se o uzavření postarat prográmátor.

5.2.2 Práce s databází na sytému Windows

Pro použití na operačním systému Windows jsou zapotřebí tři knihovny. Knihovny SQLite a Spatialite, přeložené ve formě DLL souborů. Přeložené DLL soubory, stejně tak jako zdrojové soubory knihovny, lze stáhnout z oficiálních stránek Spatialite [37]. Třetí knihovnou je SQLite JDBC Driver [40] pro přístup k databázi. Knihovna dovoluje prográmátorovi využívat pro přístup třídy definované ve standardním Java JDBC API [26].

Připojení k databázi

Jak vidíme na obrázku 5.5, připojení k databázi je složitější než u systému Android 5.2.1. V první řadě je nutné pomocí class loaderu načíst driver z knihovny SQLite JDBC Driver. Pokud se nepodaří driver načíst, je vyhozena výjimka. Druhým krokem je získání instance třídy `Connection`, v rámci které je udržováno spojení s databází. Při připojení k databázi je nutné v rámci konfigurace povolit načítání rozšíření, bez toho nebude možné používat Spatialite.

V tuto chvíli je možné provádět dotazy nad databází, ovšem databáze nezná prosotové datové typy ani prostorové operace. Abychom byli schopni pracovat s prostorovými objekty, je nutné načíst rozšíření Spatialite. To je provedeno za pomoci SQL dotazu, v jehož těle je volání příslušné funkce databáze.

Provádění dotazů

K vykonávání dotazů slouží objekt třídy `PreparedStatement`. Tento objekt reprezentuje předkompilovaný dotaz. V rámci dotazu můžeme používat parametry, v tom případě je hodnota nahrazena otazníkem. Metody pro nastavení hodnot parametrů mají prefix `set`, následovaný jménem datového typu hodnoty, například `setString(pos, value)`. Metody očekávají dva parametry, pozici a hodnotu. Narozdíl od třídy `Stmt`, jsou zde pozice parametrů číslovány od 0.

Pokud dotaz vrací data z databáze, jsou uložena v objektu třídy `ResultSet`. Tento objekt udržuje kurzor ukazující na aktuální řádek dat, která jsou výsledkem provedeného dotazu. Třída `ResultSet` poskytuje metody s prefixem `get`, pro získání hodnot jednotlivých datový typů, například `getString(name)`. Parametrem funkce je název sloupce, ze kterého chceme hodnotu získat. Alternativní možností je získat hodnotu pomocí pozice sloupce. Pomocí metody `next()` se kurzor posune na další řádek. Metoda vratí hodnotu `true` pokud další řádek existuje, `false` v opačném případě.

Po zpracování dotazu a jeho výsledku by měly být objekty tříd `PraparedStatement` i `ResultSet` uzavřeny, kvůli uvolnění zdrojů. Protože oba objekty implementují rozhraní

```
try {
  Class.forName("org.sqlite.JDBC");
} catch (ClassNotFoundException e1) {
  // nepodařilo se načíst driver
}

try {
  //vytvoření konfigurace a~povolení rozšíření
  SQLiteConfig config = new SQLiteConfig();
  config.enableLoadExtension(true);

  Connection conn = DriverManager.getConnection(;
    "jdbc:sqlite:"+dbName+".sqlite", config.toProperties());

  //provedení dotazu s~funkcí pro načtení rozšíření Spatialite
  Statement stmt = conn.createStatement();
  stmt.execute("SELECT load_extension('lib/spatialite')");
  stmt.close();
}
catch(SQLException e)
{
  //chyba při provádění dotazu
}
```

Obrázek 5.5: Připojení k databázi na systému Windows.

`AutoCloseable` [9]. Lze je vytvořit zdroj v rámci bloku `try-with-resources`, tyto objekty jsou poté uzavřeny automaticky. Tato funkce je v jazyce Java dostupná až od verze 1.7. Pro starší verze, nebo pokud programátor chce objekty uzavírat ručně, obě třídy disponují metodou `close()`.

```
try (PreparedStatement ps = connection.prepareStatement(
  ``SELECT ulice FROM domy WHERE barva = ?'')) {
  ps.setString(0, ``modra'');
  try (ResultSet rs = ps.executeQuery()) {
    while (rs.next()) {
      //tisk ulice vybraného domu
      System.out.println(rs.getString(``barva''));
    }
  }
} catch (JSQLiteException e) {
  //chyba při provádění dotazu
}
```

Obrázek 5.6: Provedení dotazu na systému Windows.

5.3 Implementace elementární procesů

Při implementaci elementárních procesů jsou v rámci aplikace mGISmobile využita vlákna pro práci s databází. Operace prováděné nad databází mohou být jak časově, tak zdrojově velmi náročné a po dobu provádění operace by uživatel nemohl s aplikací pracovat. Zároveň je u Android aplikací nutné zajistit, aby hlavní vlákno aplikace nebylo zatěžováno výpočetně náročnými operacemi. Pokud hlavní vlákno aplikace určitou dobu nereaguje na systémová volání, systém usoudí, že aplikace přestala pracovat. V takovém případě systém uživateli oznámí, že aplikace neraguje a umožní mu ji vypnout.

Pro jednoduché použití vlákem je na systému Android k dispozici třída AsyncTask [3]. AsyncTask je abstraktní generická třída, u které potomek musí implementovat metodu `doInBackground`. Kód napsaný v rámci této metody je po spuštění proveden v novém vlákně. Na obrázku 5.7 vidíme jako příklad třídu `SelectObjectWithinDistance`. Tato operace je spuštěna při výběru geoobjektu pro editaci. Na základě dotazu do databáze se určí, zda uživatel klikl na některý z geoobjektů vrstvy.

5.4 Implementace komunikace

Komunikace mezi aplikacemi ja implementována na základě standardu WFS pro přenos vektorových dat (kapitola 2.9). Aplikace implementují následující operace definované standardem WFS:

- GetCapabilities,
- DescribeFeatureType,
- GetFeature,
- InsertFeature.

Aplikace tak splňují požadavky pro službu typu Basic WFS, ovšem ne to pro typ Transactional WFS. Jednotlivé zprávy jsou implementovány dle základních požadavků a aplikace nepodporují volitelné parametry a omezující podmínky pro výběr objektů.

Nové i editované objekty jsou na server odesílány v rámci transakce `InsertFeature`. Server následně provádí insert nebo update v rámci databáze, v závislosti na tom, zda objekt již existuje.

Na obrázku 5.8 vidíme příklad metody, která vytváří v aplikaci mGISserver odpověď na operaci `GetCapabilites`. Při vytváření zprávy k odeslání pracují se obě aplikace se daty zprávy jako obyčejným textovým řetězcem. Pomocí třídy `StringBuilder` jsou do výstupního řetězce postupně přidávány jednotlivé části zprávy a po jejím dokončení je zpráva odeslána.

Při zpracování přijaté zprávy ovšem aplikace musí rozlišovat, o jaký typ operace se jedná, a dle toho data správně interpretovat. V tuto chvíli aplikace s daty zprávy pracují jako s XML dokumentem. Data jsou zpracována metodou *pull parsing*, kdy se postupně procházejí jednotlivé elementy dat a na základě přítomných elementů se rozhodne, jak budou data zpracována. Na obrázku 5.9 vidíme příklad zpracování příchozí zprávy v aplikaci mGISserver. Aplikace zjistí typ kořenového elementu, na jeho základě je určen typ požadavku a zpracování je předáno příslušné metodě implementující parsování dané zprávy.

```
private class SelectObjectWithinDistance extends
  AsyncTask<GeoObject, Void, GeoObject> {
  private MapLayer layer;
  private double distance;
  private boolean selectDeleted;

  public SelectObjectWithinDistance(MapLayer layer,
    double distance, boolean selectDeleted) {
    this.layer = layer;
    this.distance = distance;
    this.selectDeleted = selectDeleted;
  }

  @Override
  protected GeoObject doInBackground(GeoObject... params) {
    //tento kód je vykonán v~novém vlákně
    try {
      return dbWrap.selectObject(params[0], layer, distance, selectDeleted);
    } catch (DataLibraryException e) {
      return null;
    }
  }

  @Override
  protected void onPostExecute(GeoObject result) {
    //tento kód je proveden v~hlavním vlákně aplikace po ukončení
    //metody doInBackground

    //nalezení vybraného objektu v~rámci datové vrstvy
    nullSelectedObject();
    for (GeoObject o:mapLayers.get(activeLayer).getObjects()) {
      if (o.equals(result)) {
        selectedObject = o;
        break;
      }
    }

    notifyActivities(Operation.SELECT);
    super.onPostExecute(result);
  }

}

//spuštění operace
new SelectObjectWithinDistance(layer, distance, false).execute(coordinates);
```

Obrázek 5.7: Provedení dotazui na systému Windows.

5.5 Práce s GPS

Získávání polohy funguje na principu návrhového vzoru Observer. Aplikace se zaregistruje jako posluchač a systémová služba následně při získání nové polohy informuje aplikaci.

```
public String createCapablilities(List<String> layers) {
  StringBuilder sb = new StringBuilder();
  //pridani root elementu
  sb.append("<WFS_Capabilities xmlns=\"http://www.opengis.net/wfs\"
    xmlns:xsi=\"http://www.w3.org/2001/XMLSchema-instance\"
    version=\"1.0.0\">");
  //pridani pevne definovanych sekci
  sb.append(responseSections.getServiceSection());
  sb.append(responseSections.getCapabilitySection());
  sb.append("<FeatureTypeList>");
  sb.append(responseSections.getOperations());
  //pridani elementu pro jednotlive vrstvy
  for (String layer:layers) {
    sb.append("<FeatureType>");
    sb.append("<Name>");
    sb.append(layer);
    sb.append("</Name>");
    sb.append("<SRS>EPSG:4326</SRS>");
    sb.append("</FeatureType>");
  }
  sb.append("</FeatureTypeList>");
  //ukonceni root elementu
  sb.append("</WFS_Capabilities>");
  return sb.toString();
}
```

Obrázek 5.8: Vytvoření odpovědi na operaci `GetCapabilities`.

Registrace posluchače je znázorněna na obrázku 5.10. Z prostředí aplikace získáme objekt třídy LocationManager, který poskytuje přístup k lokalizačním službám na systému Android. Následně zaregistrujeme třídu implementující intefrace LocationListener (obrázek 5.11) pro získávání údajů o poloze.

```
public processReques(InputStream data) throws ParseException {
  XMLStreamReader reader = factory.createXMLStreamReader(data);

  while (reader.hasNext()) {
      int event = reader.next();

      switch (event) {
        case XMLStreamConstants.START_ELEMENT:
          switch (reader.getLocalName()) {
            case "GetCapabilities":
              return processGetCapabilities(reader);
              break;
            case "DescribeFeatureType":
              return processDescribeFeatureType(reader);
              break;
            case "GetFeature":
              return processGetFeature(reader);
              break;
            case "Transaction":
              return processTransaction(reader);
              break;
          }

          break;

        case XMLStreamConstants.CHARACTERS:
          throw new ParseException("Chybny format XML");
          break;

        case XMLStreamConstants.END_ELEMENT:
          throw new ParseException("Chybny format XML");
          break;
      }
  }
}
```

Obrázek 5.9: Zpracování příchozích dat.

```
//získání objektu pro přístup k~lokalizačním službám
locationManager = (LocationManager) context.getSystemService(
  Service.LOCATION_SERVICE);
//kontrola zda je GPS zapnuta
isGPSEnabled = locationManager.isProviderEnabled(
  LocationManager.GPS_PROVIDER);

if (!isGPSEnabled) {
  //zobrazení upozornění na zapnutí GPS
  showSettingsDialog();
}
//registrace listeneru pro získávání údajů o~poloze
locationManager.requestLocationUpdates(LocationManager.GPS_PROVIDER,
  MIN_TIME_BW_UPDATES, MIN_DISTANCE_CHANGE_FOR_UPDATES, listener);
//registrace listeneru pro získávání informací o~stavu GPS
locationManager.addGpsStatusListener(gpsListener);
}
```

Obrázek 5.10: Práce s lokalizační služnou na systému Android

```
@Override
public void onLocationChanged(Location location) {
  //metoda je volána při změně pozice
  //auktuální pozice v~modulu mobileDataLayer je aktualizována
  dl.setUserLocation(location);
}

@Override
public void onStatusChanged(String provider, int status,
  Bundle extras) {
  //metoda je volána při změně stavu GPS
}

@Override
public void onProviderEnabled(String provider) {
  //metoda je zavolána při prvním zjištění polohy
}

@Override
public void onProviderDisabled(String provider) {
  //metoda je zavolána při nedostupné GPS
}
```

Obrázek 5.11: Interface LocationListener

Kapitola 6

Protokol o mapování v terénu

Jako oblast pro vytvoření vektorových mapových vrstvev jsem zvolil Park – Obilní trh v Brně. Na obrázku 6.1 vidíme mapu parku, která byla použita jako podkladová vrstva. Rozhodl jsem se park zmapovat s ohledem pro potřeby města. Zaměřil jsem se na zaznamenání následujících objektů:

- Objekty nacházející se v parku – stromy, lavičky, lampy pouličního osvětlení, atd.
- Cesty procházející parkem.
- Celková parcela parku
- Jednotlivé travnaté plochy.

Obrázek 6.1: Mapa Park – Obilní trh.

6.1 Průběh mapování

Při mapování jsem využil mobilní telefon LG Nexus 5 s aplikací mGISmobile vytvořenou v rámci této práce. Geografickou polohu jednotlivých bodů aplikace získávala pomocí interního GPS modulu. Při mapování jsem nevyužil žádné dopravní prostředky.

V rámci přípravy jsem si vytvořil následují mapové vrstvy a jejich atributy:

- MestskeParky (polygonová vrstva):
  - Nazev – název městského parku (textový atribut).
- TravnatePlochy (polygonová vrstva).
- OstatniObjekty (polygonová vrstva):
  - Nazev – název objektu (textový atribut).
- Cesty (liniová vrstva):
  - Povrch – povrch cesty (textový atribut).
- Lavicky (bodová vrstva):
  - Stav – stav lavičky (textový atribut).
  - Material – materiál, ze kterého je lavička vyrobena (textový atribut).
- Osvetleni (bodová vrstva).
- Stromy (bodová vrstva):
  - Druh – druh dřeviny (textový atribut).
  - VekovaTridal – třída označující přibližné stáří stromu (celočíselný atribut).

Na základě mapy parku jsem si předem naplánoval přibližnou trasu při mapování. Jak můžeme vidět na mapě parku (obrázek 6.2), je park rozdělen cestami na 12 samostaných ploch. Rozhodl jsem park mapovat postupně po jednotlivých travnatých plochách. Na obrázku vlevo jsou jednotlivé plochy očíslovaný v pořadí v jakém jsem procházel parkem. Vpravo je pak naznačen postup při mapování první plochy.

Obrázek 6.2: Postup při mapování parku.

Při mapování jednotlivých oblastí jsem využil funkce paralelního mapování a postupoval jsem podobně jako v příkladě v kapitole 4.4.1. V první řadě jsem si zaznamenal souřadnice dvou okrajů plochy 1 a 2. V dalším kroku jsem se pohyboval uvnitř travnaté plochy a zaznamenával polohu jednotlivých stromů, pouličního osvětlení a laviček nacházejících se na okraji plochy. Tímto postupem jsem se postupně přesunul na protější stranu plochy a dokončil její geometrii zaznamenaním bodů 3 a 4. Obdobným způsobem jsem postupoval při mapování všech 12 částí parku.

Cesty procházejí parkem jsem do mapové vrstvy zanesl jako linie procházející celým parkem. V mapově vrstvě tedy existují 3 cesty vedoucí od severu na jih a 3 cesty vedoucí ze západu na východ. Protože v rámci jedné vrstvy může být vytvářen v jednu chvíli pouze jeden geoobjekt, nemohl jsem při průchodu parkem postupně zaznamenávat body všech cest. Jako první jsem tedy při průchodu parkem vytvářel cesty vedoucí od severu na jih. Jejich body jsem zaznamenával postupně při mezi mapováním jednotlivých travnatých ploch. Po zmapování všech travnatých ploch jsem zanzamenal zbývající cesty vedoucí za západu na východ.

Nakonec jsem se zaměřil na zaznamenání geometrie celého parku. K tomuto účelu jsem využil funkci automatického přidávání bodů. Funkci jsem nastavil tak, aby přidávala bod každých 30 sekund, v rozích parku jsem pak pozastavil, dokud nedošlo k přidání bodu. Tímto způsobem jsem zaznamenal celý obvod parku.

Na sérii obrázků 6.3 vidíme průběh mapování první travnaté plochy a postupné vytváření geoobjektů ve vrstvách.

6.2 výsledná mapa

Data zaznamenaná pomocí aplikace mGISmobile byla exportována do souborů formátu shapefile. Pro každou z výše zmíněných mapových vrstev byl vytvořen samostaný soubor. Export byl proveden za pomoci nástroje dostupného v rámci databázového rozšíření Spatialite. Exportované vrtvy byly následně vloženy do desktopové aplikace QGIS. Na obrázku 6.4 vidíme výslednou mapu, vytvořenou v programu QGIS, položenou na použité podkladové vrstvě.

6.3 Shrnutí mapování

Proběhlo mapování v lokalite Park – Obilní trh v Brně. Jako podkladová rastrová vrstva při mapování byly použity mapy dostupné na OpenStreetMap. V rámci mapování byly vytvořeny následující vektorové mapové vrstvy: MestkeParky, TranvnatePlochy, OstatniObjekty, Cesty, Lavicky, Osvetleni a Stromy. Výsledné mapové vrstvy jsou umístěny v souborech formátu shapefile na přiloženém paměťovém médiu této diplomové práce.

Obrázek 6.3: Časový průběh mapování parku.

Obrázek 6.4: Výsledná mapa parku v programu QGIS.

Kapitola 7

Závěr

V rámci této diplomové práce byl navrhnut a implementován soubor aplikací mGIS. Soubor mGIS se skládá ze dvou aplikací mGISmobile a mGISserver. Aplikace mGISmobile byla vytvořena pro mobilní zařízení a slouží ke sběru geografických dat pro GIS. Aplikace mGISserver byla vytvořena pro PC jako komunikační a databázový server. Vzájemná komunikace aplikací je založena na standartu WFS pro přenos vektorových dat.

Mezi silné strány vytvořených aplikací patří podpora práce v offline režimu a podpora paralelního mapování. Zvláště funkce paralelního mapování je při mapování přínosem. Jako slabou stránku bych ovšem označil způsob přepínání aktivních vrstev. Při paralelním mapování je změna aktivní vrstvy druhou nejčastější činnosti, po zaznamenávání bodů, a proto by tato funkce měla být lehce dostupná. V tuto chvíli je ovšem přepínání vrstvev zbytečně zdlouhavé.

V porovnání a aktuálně dostupnými aplikace znamenají aplikace mGIS posun vpřed. Většina dostupných aplikací nabízí pouze základní funkce pro práci s geometrií a postrádá přítomnost offline režimu. Naleznou se však výjimky, které aplikace mGIS svou funkčností značně převyšují, příkladem může být apliakace pcMapper.

Pro případné další navazující práce se nabízí možnost rozšíření funkcí aplikace o podporu složitějších typů geometrie, jako jsou například multipolygony. Dalším možným rozšířením může být implementace kompletní specifikace zpráv služby WFS.

Literatura

[1] Běhovský, M.; Jedlička, K.: Úvod do geografických informačních systémů. http://gis.zcu.cz/studium/ugi/e-skripta/ugi.pdf, přednáškové texty.

[2] Dokumentace: Activity. http://developer.android.com/guide/components/activities.html, [cit. 2015-05-23].

[3] Dokumentace: AsyncTack. http://developer.android.com/reference/android/os/AsyncTask.html, [cit. 2015-05-23].

[4] Dokumentace: Data File Header Structure for the dBASE Version 7 Table File. http://www.dbase.com/KnowledgeBase/int/db7_file_fmt.htm, [cit. 2015-04-11].

[5] Dokumentace: ESRI Shapefile Technical Description. http://www.esri.com/library/whitepapers/pdfs/shapefile.pdf, [cit. 2015-04-11].

[6] Dokumentace: Geography Markup Language. http://www.opengeospatial.org/standards/gml Geography Markup Language, [cit. 2015-05-23].

[7] Dokumentace: GeoTIFF - A standard image file format for GIS applications. http://www.gisdevelopment.net/technology/ip/mi03117pf.htm, [cit. 2014-11-06].

[8] Dokumentace: Gestures. http://developer.android.com/design/patterns/gestures.html, [cit. 2015-05-23].

[9] Dokumentace: Interface AutoCloseable. http://docs.oracle.com/javase/7/docs/api/java/lang/AutoCloseable.html, [cit. 2015-05-21].

[10] Dokumentace: TIFF File Format Summary. http://www.fileformat.info/format/tiff/egff.htm, [cit. 2014-11-06].

[11] Gymnázium Hranice: Kartografická zobrazení. http://gymnaziumhranice.cz/soubory/projekty/zemepis/Z-prezentace22.ppt, [cit. 2015-05-04]. studijní materiál.

[12] Hrubý, M.: Geogracké Informační systémy (předmět GIS na VUT FIT v Brně). http://perchta.fit.vutbr.cz/vyuka-gis/uploads/1/GIS-final2.pdf, 2006-12-05, studijní opora.

[13] Longley, P. A.; Goodchild, M.; Maguire, D. J.; aj.: *Geographic Information Systems and Science*. Wiley Publishing, druhé vydání, 2005, ISBN 0470870001.

[14] Rapant, P.: Úvod do geografických informačních systémů. http://gis.vsb.cz/dokumenty/ugis, 2002, skripta PGS.

[15] WWW stránky: Android NDK. https://developer.android.com/tools/sdk/ndk/index.html, [cit. 2015-05-21].

[16] WWW stránky: Apache Tomcat. http://tomcat.apache.org, [cit. 2015-01-06].

[17] WWW stránky: ArcGIS App for Smartphones and Tablets. http://www.esri.com/software/arcgis/arcgis-app-for-smartphones-and-tablets, [cit. 2014-11-10].

[18] WWW stránky: ArcGIS for Windows Mobile. http://www.esri.com/software/arcgis/arcgismobile, [cit. 2015-05-16].

[19] WWW stránky: ArcPad. http://www.esri.com/software/arcgis/arcpad, [cit. 2014-11-16].

[20] WWW stránky: Digitalpreservation. http://www.digitalpreservation.gov/formats/fdd/fdd000280.shtml, [cit. 2015-05-23].

[21] WWW stránky: ESRI Shapefile. http://www.digitalpreservation.gov/formats/fdd/fdd000280.shtml, [cit. 2015-04-11].

[22] WWW stránky: Google Maps Android API v2. http://developer.android.com/google/play-services/maps.html , [cit. 2015-02-06].

[23] WWW stránky: Google Maps/Google Earth APIs Terms of Service. https://developers.google.com/maps/terms, [cit. 2015-02-06].

[24] WWW stránky: GvSIG mobile. http://www.gvsig.org/plone/projects/gvsig-mobile, [cit. 2014-11-10].

[25] WWW stránky: GvSIG Mobile Pilot 0.3. http://www.gvsig.org/plone/projects/gvsig-mobile/official/piloto-gvsig-mobile-0.3, [cit. 2014-11-13].

[26] WWW stránky: Java JDBC API. http://docs.oracle.com/javase/8/docs/technotes/guides/jdbc/, [cit. 2015-05-21].

[27] WWW stránky: Keyhole Markup Language. https://developers.google.com/kml/documentation/?csw=1, [cit. 2015-05-13].

[28] WWW stránky: Klady a zápory WMS.
http://geo3.fsv.cvut.cz/wms/index.php?menu=wmsklaza, [cit. 2015-05-06].

[29] WWW stránky: Mapové zobrazení.
http://cs.wikipedia.org/wiki/Mapov%C3%A9_zobrazen%C3%AD, [cit. 2015-05-16].

[30] WWW stránky: Mobile apps vs. PB desktop apps.
http://www.appeon.com/support/documents/appeon_online_help/2.0/development_guidelines/ch02s01.html, [cit. 2015-02-16].

[31] WWW stránky: Open Data Commons Open Database License (ODbL).
http://opendatacommons.org/licenses/odbl/ , [cit. 2015-03-09].

[32] WWW stránky: OSMdroid. https://github.com/osmdroid/osmdroid , [cit. 2015-05-23].

[33] WWW stránky: PcMapper Db.
https://play.google.com/store/apps/details?id=com.kartdata.pcmdb&hl=cs, [cit. 2014-11-13].

[34] WWW stránky: pcMapper Lite.
https://play.google.com/store/apps/details?id=com.kartdata.pcmlite&hl=cs, [cit. 2014-11-13].

[35] WWW stránky: Shapefile over Map.
https://play.google.com/store/apps/details?id=com.dabebro.som, [cit. 2014-11-10].

[36] WWW stránky: SHP Viewer.
https://play.google.com/store/apps/details?id=com.afanche.android.ATViewSHP , [cit. 2014-11-10].

[37] WWW stránky: Spatialite. http://www.gaia-gis.it/gaia-sins/, [cit. 2015-05-21].

[38] WWW stránky: Spatialite-Android.
https://www.gaia-gis.it/fossil/libspatialite/wiki?name=splite-android, [cit. 2015-05-21].

[39] WWW stránky: SQLite. http://www.sqlite.org/ , [cit. 2015-05-23].

[40] WWW stránky: SQLite JDBC Driver.
https://bitbucket.org/xerial/sqlite-jdbc, [cit. 2015-05-21].

[41] WWW stránky: Tile servers.
http://wiki.openstreetmap.org/wiki/Tile_servers, [cit. 2015-03-09].

[42] WWW stránky: Web Feature Service.
http://www.opengeospatial.org/standards/wfs, [cit. 2015-05-04].

[43] WWW stránky: Web Map Service.
http://www.opengeospatial.org/standards/wms, [cit. 2015-05-04].

[44] WWW stránky: WMS typy dotazů.
http://geo3.fsv.cvut.cz/wms/index.php?menu=wmsdotaz, [cit. 2015-05-06].

[45] WWW stránky: World Geodetic System 1984.
http://www.unoosa.org/pdf/icg/2012/template/WGS_84.pdf, [cit. 2015-05-04].

Příloha A

Obsah CD

- Diplomová práce – zdrojové kódy práce v LaTeXu
- Soubor aplikací mGIS – zdrojové kódy aplikací mGISmobile a mGISserver
- Vektorové mapové vrstvy – soubory formátu Shapefile obsahující mapové vrstvy vytvořené v rámci mapování terénu