ZÁPADOČESKÁ UNIVERZITA V PLZNI
FAKULTA APLIKOVANÝCH VĚD
KATEDRA MATEMATIKY

Diplomová práce

Analýza klientů GIS pro Správu veřejného statku města Plzně

Plzeň, 2007 Bc. Jiří Novotný

Prohlášení

Předkládám tímto k posouzení a následné obhajobě diplomovou práci zpracovanou na závěr magisterského studia na Fakultě aplikovaných věd Západočeské univerzity v Plzni. Prohlašuji, že jsem pod vedením vedoucího diplomové práce celou tuto práci vypracoval samostatně a uvedl jsem všechny použité prameny a literaturu.

V Plzni dne 25.5.2007 ……

Podpis diplomanta

Poděkování

Poděkování patří vedoucímu diplomové práce Ing. Aleši Richtrovi (SVSmP) za metodické vedení, Ing. Karlu Sladkému (SVSmP) za věcné připomínky, Mgr. Václavu Kučerovi a Ing. Stanislavu Štanglovi (oba SITmP) za poskytnutí dat a informací o současném stavu geografického informačního systému města Plzně, v neposlední řadě poděkování patří Ing. Karlu Jedličkovi (ZČU) za rady a připomínky při zpracování diplomové práce.

Abstrakt

Projekt popisuje analýzu geografického informačního systému města Plzně a posuzuje současný stav jeho klientů používaných na Správě veřejného statku města Plzně (SVSmP). SVSmP je jedním z hlavních poskytovatelů dat pro geografický informační systém města Plzně. Projekt využívá SWOT analýzu současného stavu, která určuje silné a slabé stránky systému užití klientů GIS. SWOT analýza je tvořena z empirických údajů. Údaje jsou získány z dotazníkové akce mezi zaměstnanci a z kolektivní diskuse s vedením organizace. Pro současný stav systému užití klientů GIS je vytvořena metodika vývoje systému, ve které je systém modelován pomocí jazyka UML. Modelovací jazyk UML nabízí různé pohledy na analyzovaný systém. Odlišné pohledy umožňují efektivní vývoj systému, který je v diplomové práci ukázán na příkladu evidence veřejného osvětlení. Výsledek je jednoduchý klient GIS, ve kterém technici veřejného osvětlení aktualizují atributová data. Při této příležitosti byl vytvořen nový relační datový model podle jednotlivých požadavků techniků. Projekt je důkazem, že je-li kladen důraz na analytickou činnost, potom výsledek je jednoduché a přitom velmi efektivní řešení.

Klíčová slova

GIS, klient GIS, modelování, Správa veřejného statku města Plzně, SWOT analýza, UML.

Abstract

The project describes the analysis of the geographical information system of the city of Pilsen and analyses actual client's position used in the Public amenities authority of the city of Pilsen. The Public amenities authority of the city of Pilsen is one of the main providers for this geographical information system. The project uses the SWOT analysis of the current state, which determines strengths and weaknesses of the system use the GIS clients. The SWOT analysis is built from empirical data, whose are obtained from the questionnaire in the midst of staff and from the discussion with the board of management. For actual client's position is built the methodology of the system development, in which the system is simulated through the medium of the Unified Modeling Language. UML offers various views to the analysed system. Various views enable the useful system development, which is shown to the example of the lighting equipment department filing in this thesis. As the result is the simple GIS client, in which technicians of the lighting equipment department update attributes. At this opportunity the new relational model was built according to requirements of technicians. The project is the proof, that if the special accent is asked on the analytical operation, then the result is simple and useful answer.

Keywords

GIS, GIS client, modelling, Public amenities authority of the city of Pilsen, SWOT analysis, UML.

Obsah

Seznam obrázků v textu 8

Seznam tabulek v textu 9

1.ÚVOD 10

1.1 Informační společnost 10

1.2 Cíl diplomové práce 11

1.3 GIS města Plzně 12

1.4 Technologie a klienti GIS na SVSmP 17
1.4.1 GS Web 17
1.4.2 GSHTML 18
1.4.3 WMS klient pro MicroStation 18
1.4.4 GS Průzkumník 20
1.4.5 Bentley GeoStore 20
1.4.6 GeoStore V6 20
1.4.7 PocketGS 21

1.5 Uživatelé GIS na SVSmP 21

1.6 SWOT analýza současného stavu 22
1.6.1 Cíle SWOT analýzy 22
1.6.2 Zdroje 23
1.6.3 SWOT analýza 23

2. Metodika vývoje systému užití klientů GIS 24

2.1 Modelování 26
2.1.1 Co je to UML? 28

2.2 Požadavky 29
2.2.1 Funkční a nefunkční požadavky 29

2.3 Analýza a logická úroveň modelu 30
2.3.1 Případy užití a účastnící 31
2.3.2 Stavový diagram 34

2.4 Návrh a fyzická úroveň modelu 35
2.4.1 Diagram objektových tříd 36
2.4.2 Podstavy stavového diagramu 37
2.4.3 Diagram grafického uživatelského rozhraní 38

2.5 Implementace 40
2.5.1 Sekvenční diagram 40
2.5.2 Diagram spolupráce 42
2.5.3 Diagram činností 43
2.5.4 Diagram komponent a jejich nasazení 46

3. Vývoj systému užití klientů GIS. 47

3.1 Vývoj systému pomocí UML 47
3.1.1 Funkční požadavky 47
3.1.2 Nefunkční požadavky 48
3.1.3 Diagram případu užití 48
3.1.4 Stavový diagram systému 50
3.1.5 Diagram objektových tříd 50
3.1.6 Mapování tříd na tabulky 51
3.1.7 Podstavy stavového diagramu 54
3.1.8 Diagram grafického uživatelského rozhraní 54
3.1.9 Diagram spolupráce 57
3.1.10 Diagram činnosti 58
3.1.12 Diagram komponent 59
3.2 Diskuze 60
3.2.1 Tvorba a testování GUI 60

4. Závěr 61

5. Slovník použitých termínů a zkratek 62

6. Zdroj informací 64

7. Příloha A: Anketa a zpracování 66

8. Příloha B: Skript SQL pro vytvoření a naplnění databáze 67

9. Příloha C: Ukázka návrhu jednoduchého klienta GIS 70

Seznam obrázků v textu

Obr. 1.1: Schéma vazeb GIS (převzato z [16]). 14
Obr. 1.2: Třívrstvá architektura. 16
Obr. 1.3: Příklad webové mapové služby WMS (převzato z [19]). 19
Obr. 1.4: Využívání aplikace GS Web (převzato z [16]). 22
Obr. 1.5: SWOT analýza současného stavu. 23
Obr. 2.1: Popis a vývoj systému (vytvořeno ve spolupráci s Ing. Karlem Jedličkou). 25
Obr. 2.2: Množství vykonané práce v závislosti pracovním procesu podle [1]. 29
Obr. 2.3: Množství vykonané práce v pracovním procesu analýza podle [1]. 30
Obr. 2.4: Případ užití existujícího systému. 32
Obr. 2.5: Stavy a přechody grafického uživatelského rozhraní pro všechny aktéry. 34
Obr. 2.6: Množství vykonané práce v pracovním procesu návrh podle [1]. 35
Obr. 2.7: Ukázka diagramu objektových tříd systému veřejného osvětlení. 36
Obr. 2.8: Podstavy ve stavu práce pro aktéra pracovník EM v ZadatData. 37
Obr. 2.9: Podstavy ve stavu práce pro aktéra pracovník EM v SmazatData. 37
Obr. 2.10: Podstavy ve stavu práce pro aktéra pracovník EM v AktualizovatData. 38
Obr. 2.11: Podstavy ve stavu práce pro aktéra uživatel GIS v ZobrazitData. 38
Obr. 2.12: Diagram toku popisující GUI pro příklad inicializace. 39
Obr. 2.13: Prototyp GUI pro příklad inicializace. 40
Obr. 2.14: Množství vykonané práce v závislosti pracovním procesu podle [1]. 40
Obr. 2.15: Sekvenční diagram zachycující všechny případy užití aktéra pracovník EM. 41
Obr. 2.16: Sekvenční diagram zachycující případ užití aktéra uživatel GIS. 42
Obr. 2.17: Diagram spolupráce zachycující všechny případy užití aktéra pracovník EM. 43
Obr. 2.18: Diagram spolupráce pro GUI. 43
Obr. 2.19: Diagram činností popisující práci pracovníka EM. 44
Obr. 2.20: Diagram činností popisující práci uživatele GIS. 45
Obr. 2.21: Diagram komponent a jejich nasazení. 46
Obr. 3.1: Případ užití optimalizovaného systému. 49
Obr. 3.2: Ukázka diagramu objektových tříd systému VO. 51
Obr. 3.3: Mapování dědičnosti zahrnutím podtříd do nadtříd. 52
Obr. 3.4: Mapování asociací 1:N. 53
Obr. 3.5: Výsledný relační datový model VO s číselníky. 53
Obr. 3.6: Podstavy ve stavu práce pro aktéra technik v případu užití AktualizovatData. 54
Obr. 3.7: Diagram toku popisující práci aktéra technik pro případ užití AktualizovatData. 55
Obr. 3.8: Prototyp GUI pro případ užití AktualizovatData o osvětlení pro aktéra technik. 56
Obr. 3.9: Prototyp GUI pro aktualizaci osvětlení a svítidlech pro aktéra technik. 57
Obr. 3.11: Diagram spolupráce pro aktéra technik. 57
Obr. 3.12: Diagram činností popisující práci technika. 58
Obr. 3.13: Diagram komponent a jejich nasazení v optimalizovaném systému. 59

Seznam tabulek v textu

Tab. 2.1: Detail případů užití ZadatData. 33
Tab. 2.2: Detail případů užití SmazatData. 33
Tab. 2.3: Detail případů užití AktualizovatData. 33
Tab. 2.4: Detail případů užití UložitData. 34
Tab. 2.5: Detail případů užití ZobrazitData. 34
Tab. 3.1: Detail případů užití ZobrazitData. 49

1.ÚVOD

1.1 Informační společnost

Rozvoj informační společnosti v České republice je již nezadržitelný. Rozvoj sebou přináší zvyšování hodnoty informace jako jednoho ze základních ekonomických zdrojů. Dnešní společnost lze chápat jako společnost, která využívá digitální zpracování, uchování a přenos informací. Ze zpracování, uchování a přenosu informací se stává významná ekonomická aktivita, která prostupuje tradičními ekonomickými či společenskými aktivitami a vytváří zcela nové možnosti, jenž ovlivňují charakter společnosti. Informační společnost již netvoří pouze elitu, která má možnost pracovat s informacemi. Tato elita se dnes již rozšířila směrem k plnohodnotné společnosti. Důkazem těchto řádků je fakt, že informace má dnes v některých případech větší význam než základní ekonomické zdroje jako je například půda či kapitál. Umět pracovat s informacemi je základní předpoklad pro rozvoj této společnosti. (Zpracováno podle [2]).

Nedílná součást rozvoje informační společnosti se týká rozvoje prostorové informace. Do širokého spektra oborů proniká prostorová informace především díky geografickým informačním systémům (GIS). Tyto systémy představují nejvyšší stupeň kvalitního zpracování informací, protože *"prakticky všechna data zahrnutá do informačních systémů obsahují i územní vyjádření"* [21].

Postupným přechodem z postindustriální společnosti na informační společnost dochází k zvýšení požadavků nejen na informační systémy, ale i na jejich provozovatele, kteří musí rychleji reagovat na požadavky uživatelů a tím i na rozvoj systému z hlediska jeho obsažnosti. Veřejná správa se tedy stejně jako většina komerční sféry neobejde bez kvalitních informačních technologií.

V širším kontextu můžeme správu veřejných záležitostí dělit na:

- státní správu,
- územní samosprávu,
- zájmovou samosprávu.

Územní samosprávu vykonávají obce a kraje za pomoci informačních systémů. Rozvoj informačních systémů měst byl dosud zaměřen na vnitřní problematiku úřadů. Postupně však začínají být informační systémy vnímány jako základní informační infrastruktury obce. Tyto změny jsou nejen v oblasti technické, ale i organizační, kde došlo ke změně organizace práce úřadů podle potřeb veřejnosti. Nedílnou součástí informačních systémů měst jsou právě geografické informační technologie a mapové služby. Díky provázání s dalšími agendami úřadu tak mohou přispět k vyšší efektivitě práce.

Publikace *Česká města na cestě k informační společnosti* vydaná pražským magistrátem společně se Svazem měst a obcí v srpnu 2005, uvádí výsledky průzkumu využívání informačních technologií ve statutárních městech ČR. Vybraná česká statutární města se zúčastnila celoevropského benchmarkingového projektu *eCitizenship for All – survey and award* organizace EUROCITIES Knowledge Society Forum – TeleCities a společnosti Deloitte. Z měst, která se zúčastnila průzkumu, má přes 70% vypracovanou informační strategii, kde podnětem ke vzniku strategie je zákon o informačních systémech veřejné správy (ISVS). Informační systémy měst můžeme rozdělit na aplikační oblasti pro výkon samosprávy:

- základní ekonomické agendy,
- správa majetku,
- agenda veřejné správy,
- spisová služba,
- GIS a mapové služby,
- správa webových stránek.

Z důvodu, že aplikační oblast GIS a mapové služby má před sebou velkou budoucnost, dokonce podle časopisu Nature je geoinformatika třetím nejperspektivnějším oborem [3], bude konkrétní problematice aplikační oblasti GIS a mapové služby v informačním systému města Plzně věnována tato diplomová práce.

1.2 Cíl diplomové práce

Cílem diplomové práce je:

1. Analýza GIS města Plzně.
2. Analýza systému užití klientů GIS na Správě veřejného statku města Plzně (SVSmP).
3. Metodika vývoje systému užití klientů GIS.
4. Zkvalitnění využívání geografických informací na SVSmP.

Pozn. SVSmP je příspěvková organizace spadající pod Magistrát města Plzně, kde aplikační oblast GIS a mapové služby tvoří technologie firmy GEOVAP, spol. s r.o. SVSmP tvoří v GIS města Plzeň jeden z hlavních zdrojů prostorových i atributových dat (dále geodat), které jsou poskytovány ostatním uživatelům městského informačního systému. SVSmP má ve své správě většinu majetku, který svým charakterem patří do GIS a je určen prostorově. Jedná se zejména o veřejně přístupný majetek města, především:

- místní komunikace včetně jejich příslušenství,
- mosty,
- lávky,
- podchody,
- opěrné zdi,
- městské informační a orientační systémy,
- veřejné osvětlení,
- světelná signalizační zařízení včetně kamerového dohledu,
- řadu uzavřených areálů,
- městské hájovny,
- městskou a příměstskou zeleň,
- lesy,
- vodní plochy a toky,
- pozemky a další majetek.

1.3 GIS města Plzně

Plzeň je čtvrtým největším městem v České republice. V západní části Čech zaujímá výrazné dominantní postavení jako silné průmyslové, obchodní, kulturní a správní centrum.

Pro chod takto významného města je zapotřebí kvalitní městský informační systém. Proto je kladen důraz na tvorbu a vývoj městského GIS. Důkazem je mapový portál města a ocenění získané provozovatelem a koordinátorem Správou informačních technologií města Plzně (SITmP):

- 1. místo v kategorii Města a obce na konferenci ISSS – Internet ve státní správě a samosprávě v Hradci Králové ze dne 25.3.2002.
- 3. místo v soutěži o nejlepší panel za plakát GIS a optickou síť MISNet na konferenci GIS SEČ v Chrudimi ze dne 13.6.2003
- Zvláštní cena poroty v kategorii GIS pro lepší služby veřejné správy na konferenci ISSS – Internet ve státní správě a samosprávě v Hradci Králové ze dne 4.4.2005

Podle [16], tato organizace využívá zvýšeného zájmu o prostorové informace a projevuje snahu implementovat GIS do života obyvatel města prostřednictvím www stránek města.

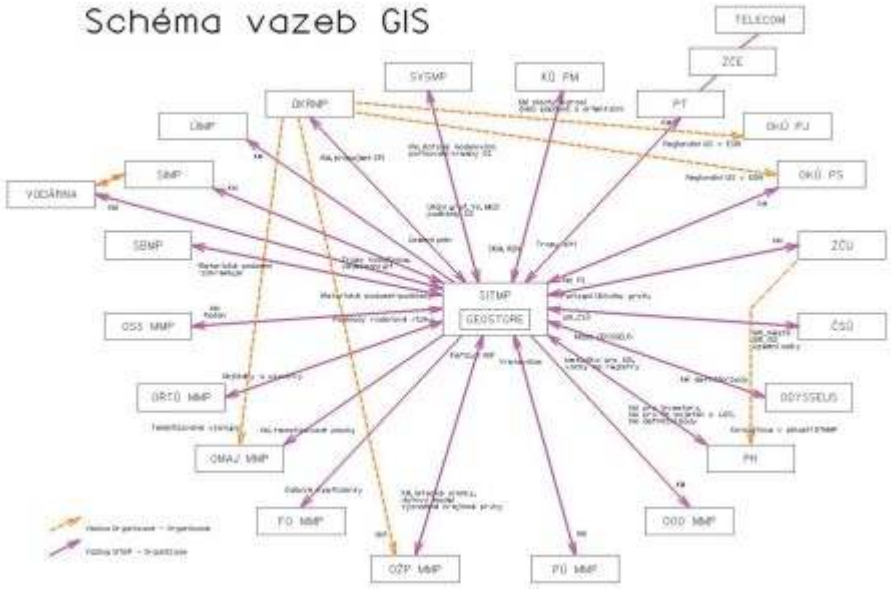
Svoji práci [16] sama definovala: *„Městský GIS chápeme jako informační systém pracující s prostorovými daty, jež se vztahují na území města. Umožňuje zobrazovat, vyhledávat a analyzovat grafické objekty spolu s jejich připojenými informacemi.“* V současné době se tato definice rozšířila nejenom na území města, ale i na okolí města Plzně.

Požadavky na funkčnost GIS zahrnují:

- Sklad geografických dat.
- Správa, úprava datového modelu a rozvíjení systému.
- Data z katastru nemovitosti.
- Aktualizace dat.

Všechna data jsou uložena v centrálním datovém skladu. Znamená to uložení dat v jednotné struktuře a na jednom místě, ke kterému se přistupuje přes jasně definované rozhraní bez nutnosti konverzí a exportů dat. Centrální datový sklad v GIS města Plzně spravuje SITmP.

SITmP se stará o rozvoj systému vlastními silami, což obnáší koordinaci subjektů města při tvorbě GIS a zajišťování metodického vedení těchto subjektů. Jednotlivé subjekty města mají odpovědnost za pořizování a aktualizaci GIS. Tyto vazby jsou vidět na obr.1.1 z roku 2002, který dokumentuje složitost vazeb, i když některé subjekty již zanikly nebo je nahradily jiné.



Obr. 1.1: Schéma vazeb GIS (převzato z [16]).

Mezi hlavní zdroj prostorových dat v městském systému patří informace z Katastrálního úřadu Plzeň-město, který dodává do systému informace o majetkoprávních vztazích na území města. Katastrální mapa je díky ucelenému rozsahu hlavním mapovým podkladem pro tento GIS.

Aktualizace všech dat probíhá on-line, což umožňuje po zadání dat okamžitou vizualizaci prostřednictvím Intranetu/Internetu. Na aktuálnosti dat se podílí více než 20 pracovišť v režimu on-line.

GIS města Plzně poskytuje prohlížení aktuálních geografických dat širokému počtu uživatelů a slouží taktéž velkému množství uživatelů v oblasti státní správy, územní i zájmové samosprávy a veřejnému sektoru. Koncoví uživatelé využívají data pouze prostřednictvím běžného webového prohlížeče, čímž je zajištěno jednoduché ovládání.

Charakteristickým rysem GIS města Plzně je zpracování geografických dat standardní relační databázovou technologií a nasazením webových prostředků na jejich šíření a využívání.

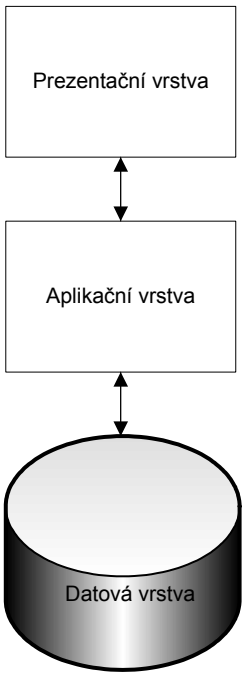
Relační databáze prezentují modelovaný svět pomocí entitně-relačního modelu, kde je základem převedení komplexních struktur modelované skutečnosti do dvourozměrných tabulek a nalezení vztahů mezi nimi. Relační databáze je sestavená z řady tabulek, jejichž klíčový atribut je vázán na atribut v jiné tabulce. Takto propojené entity jsou na sobě určitým způsobem závislé. Tyto závislosti jsou založeny na klíčových hodnotách uložených v příslušných sloupcích. Vizuální reprezentace dat pomocí entitně-relačního modelu má nezastupitelnou úlohu nejen pro návrh databázových aplikací, ale i při jejich optimalizaci a odstraňování chyb. Efektivní datový model přesně a úplně popisuje nároky zadání, zároveň je použitelný pro tvůrce databází, eliminuje redundanci dat a je nezávislý na hardwaru a softwaru.

Při složité modelové skutečnosti, se entitně-relační model stává velmi nepřehledným. Alternativou relačních databází jsou databáze objektové, se kterými se blíže seznámíme v 2. kapitole *Metodika vývoje systému užití klientů GIS*. Obecně u relačních databází je základní výhodou relativně snadná modifikace a propojování tabulek a s nimi spojená možnost dotazů. Slabým místem je nízká efektivnost zpracování.

GIS města Plzně je vystavěn v třívrstvé architektuře. Základ třívrstvé architektury podle obr. 1.2 tvoří:

- datová vrstva,
- aplikační vrstva,
- prezentační vrstva.

Na datový sklad se napojuje software pro tvorbu a aktualizaci dat.



Obr. 1.2: Třívrstvá architektura.

Technologická architektura GIS města Plzně je tvořena centrálním datovým skladem v databázové platformě Microsoft SQL Server 2000. Správu prostorových dat uložených v relační databázi zajišťuje systém GeoStore firmy GEOVAP, spol. s r.o. Prostorová data jsou v tabulkách relační databáze uložena ve speciálních sloupcích, které můžeme dohromady považovat za atribut geometrie. „*V kombinaci s léty ověřenými schopnostmi RDBMS, GeoStore tímto představuje vysoce efektivní prostředek pro vstup, správu, výstup a základní analytické zpracování prostorových dat v relačním databázovém systému*“ [4].

Aplikační vrstva využívá technologie výkonného aplikačního serveru, který přistupuje k datové vrstvě. Zdrojová data transformuje do internetového formátu a publikuje je do prostředí Intranet/Internet. Přístup k datům je zabezpečen a neumožňuje destruktivní akce ve zdrojových datech. Bezpečnost dat je zajištěna tak, že klient neobdrží originální data, ale pouze jejich komprimovaný obraz. Na straně klienta nedochází k ukládání dat a data jsou chráněna proti zcizení a zneužití. Protože systém nepřenáší data, ale pouze jejich komprimovaný obraz, dochází k redukci přenášených dat a tím i k výraznému zrychlení přenosu.

Prezentační vrstva je vytvořena tak, aby splňovala tři hlavní zásady:

- jednoduché ovládání,
- bezpečnost dat,
- rychlost přístupu k datům.

Pro uživatelský přístup k datům se používá běžný www prohlížeč, který podporuje software Java. V prostředí prohlížeče pracuje uživatel na běžné HTML stránce. Uživatel je schopen pracovat s klientem intuitivně a bez jakéhokoliv zaškolení, díky automatickému navigování uživatele klientem.

Pro pořizování, import a export geodat slouží grafický editor MicroStation firmy Bentley s připojenou nadstavbou GeoStore firmy GEOVAP, která komunikuje s databází nebo přímo nová verze programu GeoStore V6 stejné firmy, který je zároveň grafickým editorem.

1.4 Technologie a klienti GIS na SVSmP

Na SVSmP se používá pro zobrazování dat uložených v centrální databázi klient GS Web. Prostorová data se do relační databáze ukládají z prostředí grafického editoru MicroStation nebo GeoStore V6, který dokáže ukládat i atributová data. Samostatné atributové data se do relační databáze ukládají z prostředí editoru GS Průzkumník. Ve všech editorech je rovněž možné provádět jejich aktualizaci.

1.4.1 GS Web

GS Web je klient pro využívání geodat. Slouží jako prostředek pro zobrazení geodat uložených v relační databázi. Jediný software nutný pro prohlížení prostorových dat je standardní www prohlížeč, který podporuje Java. GS Web spojuje výhody prostředí Intranet/Internet a relační databáze. Dále dává možnost data uložená v databázi prezentovat kdekoli a kdykoli při zachování bezpečnosti dat. Přístup k datům pro různé uživatele lze standardně určit na straně databáze. Na straně klienta se neukládají žádná data, proto nelze prostřednictvím programu GS Web získat zobrazovaná prostorová data. GS Web je tedy jednoduchý a proti zneužití zabezpečený prostředek pro plošné šíření geografických dat. GS Web je užívaný převážně ve vnitřních organizacích města Plzně, kde klient obsahuje kompletně přístupné všechny vrstvy. (Zpracováno podle [4] a [16]).

Aplikační logika a komunikace s databází produktu GS Web je realizována jako CGI (Common Gateway Interface) aplikace v prostředí www serveru. Vlastní prezentaci prostorových dat zprostředkovává java applet, který je rovněž uložen na aplikačním serveru.

Klient umožňuje:

- Výběr objektů z databáze.
- Zapínání a vypínání zobrazovaných vrstev včetně legend.
- Změna měřítka.
- Posun v mapovém okně.
- Měření délek a ploch.
- Vyhledávání objektů.
- Přehledové tisky v měřítku.

1.4.2 GSHTML

GSHTML je klient pro využívání prostorových dat, který je určen především pro veřejnost. Slouží jako prostředek pro zobrazení geodat uložených v relační databázi. Jediný software nutný pro prohlížení prostorových dat je standardní www prohlížeč. GSHTML klient je velmi jednoduchý na ovládání. Veřejnosti jsou přístupné redukované vrstvy podle tématu. (Zpracováno podle [4] a [16]).

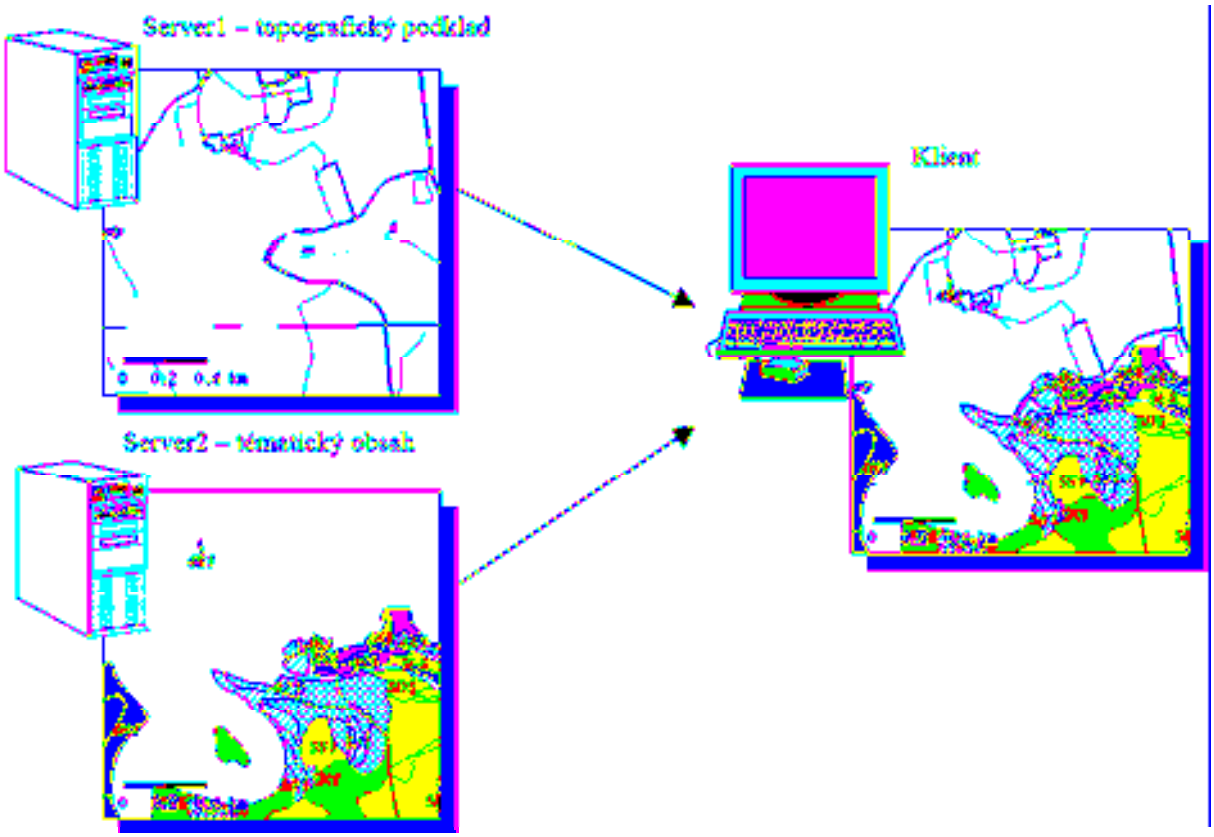
Klient umožňuje:

- Zapínání a vypínání zobrazovaných vrstev.
- Změna měřítka.
- Posun v mapovém okně.
- Zobrazování přehledové mapy.

1.4.3 WMS klient pro MicroStation

Webové služby představují další generaci internetových aplikací. Umožňují interakci zejména aplikace - aplikace v prostředí Internet. Komunikace je umožněna na základě

standardizovaných protokolů. Výhodou je, že uživatel má v jedné aplikaci přístup k mapám z několika serverů a přistupuje jen k těm službám, které za daných podmínek potřebuje. (Zpracováno podle [5]).



Obr. 1.3: Příklad webové mapové služby WMS (převzato z [19]).

WMS (Web Map Service) klient pro MicroStation je klient pro využívání prostorových dat. Tento softwarový produkt umožňuje přístup k datům poskytovaných WMS servery na Internetu. WMS byla vyvinuta konsorciem Open GIS (OGC) jako standard pro přenášení map a polohově určených dat přes Internet za účelem vytváření mapových kompozic. (Zpracováno podle [4] a [16]).

Klientská aplikace zjistí schopnosti mapového serveru, zejména jaké mapové vrstvy nabízí. Potom již může na mapový server posílat požadavky na generování obrázků s definovanými vrstvami, v definovaném souřadnicovém systému, formátu a definované velikosti. Po zadání požadavků server pošle informace.

Aplikace implementuje službu WMS do prostředí grafického editoru MicroStation. Uživatel definuje WMS zdroje, datové vrstvy a jejich zobrazení. Data jsou přenášena

standardním protokolem HTTP (Hyper Text Transfer Protocol) a připojována jako referenční rastrová vrstva.

Klient umožňuje:

- Efektivní práci s rozsáhlými datovými zdroji v prostředí Internet/ Intranet.
- Efektivně řešit integraci dat z různých zdrojů.
- Prezentovat data v rastrovém tvaru i přesto, že na serverech jsou uložena v různých formátech.

1.4.4 GS Průzkumník

GS Průzkumník je klient pro aktualizaci atributových dat. Slouží jako prostředek ke čtení a editování popisných dat uložených v databázové tabulce společně s grafickou reprezentací prvku. Aplikace může pracovat samostatně nebo ve spolupráci s programem MicroStation. Spolupráce s MicroStation umožňuje navíc zobrazení grafické reprezentace prvku a obráceně nalezení popisných údajů o prvku, na jehož grafickou podobu je ukázáno v DGN výkresu. Obdobně je umožněno propojit GS Průzkumník s prostorovým vyjádřením prvku v GS Web. (Zpracováno podle [4] a [16]).

1.4.5 Bentley GeoStore

Bentley GeoStore je klient pro aktualizaci geodat. Slouží jako prostředek pro pořizování, import a export prostorových dat. Grafický editor tvoří produkt firmy Bentley - MicroStation nebo MicroStation GeoOutlook. Klient má k dispozici specializované nástroje, které umožňují zachycovat celý životní cyklus geografických objektů (prvek nový, aktualizovaný, smazaný, historizovaný), jejich úpravu a topologické kontroly. K tomuto prostředí je připojena nadstavba GeoStore, která prostřednictvím aplikační vrstvy komunikuje s databází. (Zpracováno podle [4] a [16]).

1.4.6 GeoStore V6

GeoStore V6 je desktop GIS a zároveň grafický editor. GeoStore V6 je moderní GIS vyvinutý v technologii Microsoft .NET. Spojuje v sobě nejdůležitější funkce pro tvorbu, aktualizaci a správu geodat s pokročilými funkcemi GIS. Může sloužit jako grafický editor

s plnou škálou editačních funkcí obvyklých u CAD nástrojů nebo jako pokročilý desktop GIS. (Zpracováno podle [4] a [16]).

GeoStore V6 je programovatelný systém. Hlavní metody a datové struktury jádra systému jsou veřejné. To přináší nejvyšší stupeň otevřenosti vůči uživatelům - vývojářům. Ti mohou rozvíjet funkcionalitu systému vlastními moduly a aplikacemi vyvíjenými standardními prostředky technologie .NET.

1.4.7 PocketGS

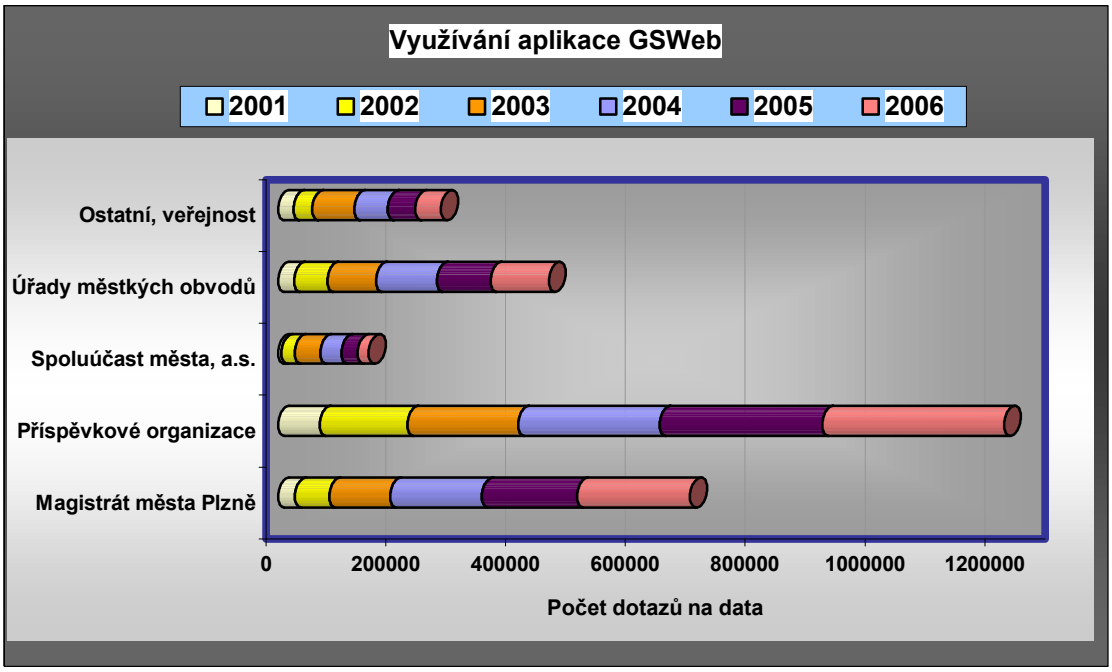
PocketGS je mobilní nadstavbou technologie GeoStore pro prezentaci vektorových, rastrových a popisných dat ve vzájemných souvislostech na mobilní zařízení PDA (Personal Digital Assistant) a lokalizace polohy podle GPS (Global Positioning System). Je provozovatelný na všech mobilních zařízeních s operačním systémem, který podporuje technologii .NET Compact Framework. PocketGS je používaný hlavně pro sběr a kontrolu dat v terénu. (Zpracováno podle [4] a [16]).

Klient umožňuje:

- Sběr dat v terénu.
- Lokalizace polohy dle GPS.
- Navigaci v terénu.
- Lokalizaci objektů GIS dle uživatelem na definovaných kritérií.
- Zobrazení popisných informací.

1.5 Uživatelé GIS na SVSmP

Koncoví uživatelé na SVSmP, která spadá pod příspěvkové organizace města využívají data s centrální databáze prostřednictvím běžného webového prohlížeče a klienta pro využívání dat GS Web. Počet uživatelů programu GS Web není omezen a v průběhu let tento počet narůstá.



Obr. 1.4: Využívání aplikace GS Web (převzato z [16]).

1.6 SWOT analýza současného stavu

SWOT (Strengths Weaknesses Opportunities Threats) analýza je komplexní metoda, kterou lze použít ke kvalitativnímu vyhodnocení veškerých relevantních stránek fungování klientů GIS na SVSmP. Je silným nástrojem pro celkový rozbor vnitřních i vnějších činitelů a v podstatě zahrnuje postupy technik strategické analýzy.

SWOT analýza v tomto projektu hodnotí silné a slabé stránky zejména se zaměřením na interní prostředí SVSmP, příležitosti a hrozby zejména se zaměřením na externí prostředí organizace. Vzájemnou interakcí silných a slabých stránek na jedné straně vůči příležitostem a hrozbám na straně druhé lze získat nové kvalitativní informace, které charakterizují a hodnotí úroveň jejich vzájemného střetu. SWOT analýza vychází z předpokladu, že strategického úspěchu ve správném fungování klientů GIS na SVSmP lze dosáhnout maximalizací silných stránek a příležitostí, minimalizací slabých stránek a hrozeb.

1.6.1 Cíle SWOT analýzy

Základní cíle SWOT analýzy jsou:

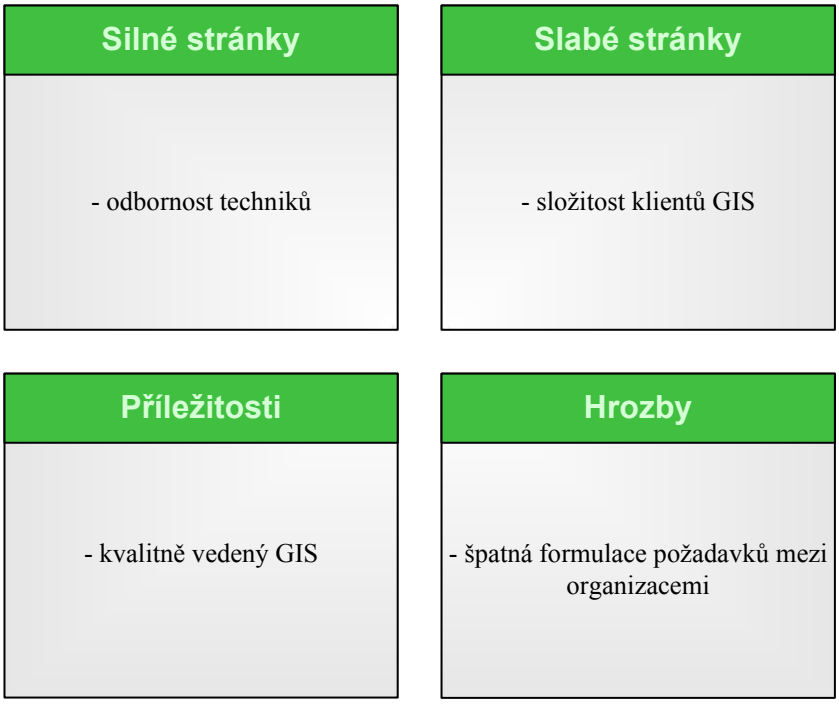
- Stanovení silných a slabých stránek v jednotlivých problémových oblastech současného stavu klientů GIS.
- Odhalit příležitosti a hrozby podle jednotlivých problémových oblastí.

1.6.2 Zdroje

SWOT analýza je tvořena z empirických údajů získaných z následujících zdrojů:

- *Z dotazníkové akce mezi zaměstnanci organizace*. Ankety se zúčastnilo 10 zaměstnanců, kteří odpovídali na otázky týkající se složitosti programového vybavení a spolupráce se SITmP. Otázky a zpracování ankety viz. Příloha A.
- *Z kolektivní diskuse s vedením organizace*. Diskuse potvrdila výsledky provedené ankety.

1.6.3 SWOT analýza



Obr. 1.5: SWOT analýza současného stavu.

Na SVSmP pracuje velké množství odborníků na spravovaný majetek. Maximalizovat silnou stránku znamená zmenšit úsilí odborníků na obsluhu programového vybavení. Důsledkem je, že odborníci nesmí být zatíženi složitými programovými aplikacemi.

Maximalizovat příležitost kvalitně vedeného GIS znamená prohlubovat spolupráci se SITmP, která je provozovatelem GIS města Plzně a vyvarovat se hrozbě odtržení informatiků od reálných problémů SVSmP. Důsledkem je, že na SVSmP musí pracovat odborníci na GIS.

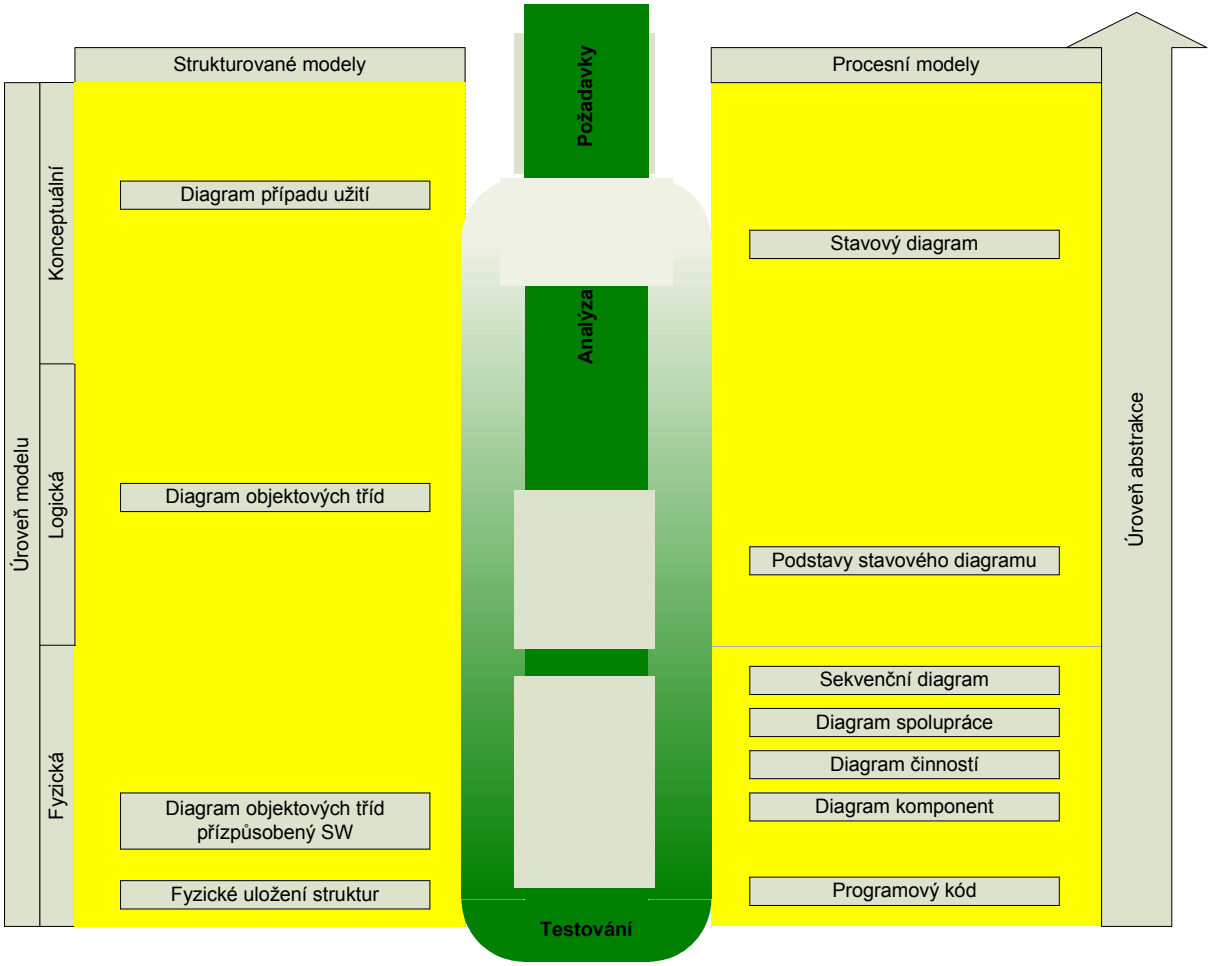
2. Metodika vývoje systému užití klientů GIS

Druhá část práce se bude zabývat teoretickou stránkou metodiky vývoje systému. Tato část bere na vědomí náročnost problematiky, a proto budou popsány jednotlivé kroky srozumitelnou formou, s množstvím diagramů, které nabízejí různé pohledy na analyzovaný systém. Seznámíme se ze základy modelovacího jazyka UML (Unified Modeling Language) na konkrétním případě evidence geodat. Bude to první představa o používání tohoto moderního a výkonného modelovacího jazyka a pochopení základních principů na praktické ukázce systému evidence veřejného osvětlení na SVSmP, který bude v následující kapitole vylepšován. Kapitola bude rozebírat krok po kroku systém evidence veřejného osvětlení a výsledným produktem tohoto úvodu do UML bude dokumentace systému evidence veřejného osvětlení na SVSmP.

V projektu se budeme držet osvědčené typografické konvence, která se používá v odborné literatuře o UML, např. [8]. *Kurzívou* budeme odlišovat názvy diagramů, tříd, aktérů, tabulek, procesů apod. `Písmem Courier` budeme odlišovat syntaxi zápisu.

Většina aplikací je dnes psána v objektově-orientovaných jazycích. Objekty popisují komplexní realitu, dovolují urychlit vývoj aplikace a zjednodušit jej díky polymorfismu a dědičnosti. Čím složitější je modelovaná skutečnost, tím spíše se ukáže, že ji lze popsat pouze objektově, nebo že entitně-relační model by byl tak složitý, až by se stal v praxi nepoužitelný. Objekty mají oproti relačním strukturám jednu zásadní nevýhodu. Na rozdíl od entitně-relačního modelu neexistuje ucelená teorie a metodika, která říká, jak objektový model navrhnout. Metodika představuje strukturu a charakter kroků vývoje systému. Metodika obsahuje pět základních pracovních postupů jež určují, co je třeba udělat, a způsob, jakým toho dosáhneme. Tyto pracovní postupy se mohou navzájem prolínat, čímž dosáhneme vylepšování jednotlivých pracovních postupů.

- *Požadavky.* Zachycují to, co by měl systém dělat.
- *Analýza.* Vybroušení požadavků a jejich strukturování.
- *Návrh.* Realizace požadavků v architektuře systému.
- *Implementace.* Tvorba fyzického modelu.
- *Testování.* Ověřování, zda implementace funguje tak, jak se od ní očekává.



Obr. 2.1: Popis a vývoj systému (vytvořeno ve spolupráci s Ing. Karlem Jedličkou).

Pracovní postupy jsou podle obr. 2.1 chápány tak, že než se začne programovat musí vývojáři znát strukturu požadavků. Tato struktura požadavků a její vybroušení tvoří ve vývoji systému *konceptuální úroveň modelu*, která může být reprezentována:

- *Diagramem případu užití.*
- *Stavovými diagramy systému.*

Díky těmto modelům můžeme provést další analytický krok, který ve vývoji systému tvoří *logickou úroveň modelu.* Logická úroveň může být reprezentována:

- *Diagramem objektových tříd.*
- *Podstavy stavového diagramu.*

Převedením těchto modelů do implementace, získáme *fyzickou úroveň modelu*, která může být reprezentována:

- *Sekvenčním diagramem.*
- *Diagramem spolupráce.*
- *Diagramem činností.*
- *Diagramem komponent.*
- *Diagramem objektových tříd.*
- *Programovým kódem.*
- *Fyzickým uložením v databázi.*

Výsledek se po otestování a nasazení aplikace stane funkčním systémem.

2.1 Modelování

Modelování lze chápat jako proces tvorby modelu, kde model je popis systému vytvořený za určitým účelem. *„Model má při tvorbě softwaru stejný význam jako výkres při realizaci stavby“* [8]. Naopak podle [15] dobře vytvořený model usnadňuje komunikaci mezi jednotlivci, kteří se podílejí na realizaci projektu, a zároveň nám pomůže zajistit, aby systém byl navržen rozumně. Dalším důvodem, proč vytváříme model je, že jako celek nemůžeme složitější systém nikdy pochopit. Chování modelu by mělo odpovídat chování systému. Míra podobnosti chování modelu a systému závisí podle [7] na:

- komplexnosti systému,
- na zvoleném jazyce popisu systému,
- na schopnostech návrháře modelu.

Abstraktiva, jako jsou procesy modelu a struktura kódu, lze samozřejmě popsat vlastními slovy, ale specifikací, konstrukcí a vizualizací vztahů usnadníme programátorům a návrhářům aplikací vzájemnou komunikaci. Nebo-li *„vymezení každého abstraktivního pojmu je možné vždy jen použitím vhodného dorozumívacího jazyka. Zvolíme-li pro tyto účely běžný hovorový jazyk, pak bude získaná teorie sice velmi nepřesná, bude však adaptabilnější ke konkrétním potřebám. Zvolíme-li jazyk matematiky, pak bude přesnost a adaptabilita teorie závislá na stupni exaktnosti této matematiky.“* [23]. Při modelování se používají nejrůznější dorozumívací jazyky, které jsou s rostoucí složitostí systémů stále potřebnějším nástrojem umožňujícím modelování a různé metody vizualizace modelů. Podle [15] musí modelovací jazyk kromě kompromisu mezi adaptabilitou ke konkrétní potřebě a přesnosti modelu ještě obsahovat:

- *Modelovací prvky.* Základní koncepty a sémantická pravidla.
- *Notaci.* Symboly pro nejrůznější prvky systému.
- *Návody.* Základní pokyny pro použití jednotlivých prvků jazyka.

Tuto potřebu splňuje právě UML (Unified Modeling Language). Zejména u rozsáhlejších řešení. Jednotný modelovací jazyk umožňuje pomocí grafických symbolů přehledně a srozumitelně dokumentovat problém a usnadňuje tak komunikaci mezi zákazníky, návrháři, programátory a jednotlivci, kteří se podílejí na realizaci projektu. UML se stal zavedeným standardem pro dokumentování ve všech fázích vývoje systému. UML plně podporuje objektový přístup a umožňuje integraci nástrojů, čímž se stává prostředkem, díky němuž mohou vývojáři vytvářet stále nové hodnoty.

Samo použití UML nezajistí projektu úspěch, ale i přesto vyřeší řadu problémů. Již byla zmíněna výhoda UML při komunikaci a také při tvorbě dokumentace. Dalšími výhodami UML je podle [15] snížení nákladů při:

- školení pracovníků,
- nákupu nových prostředků při přechodu k novým projektů,
- nákupu nových prostředků při přechodu k nové organizaci.

2.1.1 Co je to UML?

UML vytvořili metodici Grady Booch, Ivar Jacobson a Jim Rumbaough ve společnosti Rational Software. Získávali přitom informace od jiných metodiků, mnoha dodavatelů programového vybavení i od koncových uživatelů. Vývoj byl zahájen v říjnu roku 1994 a cílem bylo sjednotit různé stávající systémy do jediného nejlepšího modelovacího jazyka. V listopadu 1997 ho přijala OMG (Objekt Management Group) za standard. (Zpracováno podle [15]).

Při tvorbě UML byly podle [15] podstatné následující cíle:

- Vytvořit snadno použitelný a přehledný modelovací jazyk. Všechny základní požadavky na modelování všech systémů jsou stejné, je proto vhodné, aby modelovací jazyk přímo definoval koncepty potřebné k vytvoření základního modelu systému.
- Umožnit rozšíření základních principů. UML lze pro většinu případů použít pouze se znalostí základních principů. Speciální případy lze řešit pomocí modifikace základních postupů.
- Vytvořit modelovací jazyk nezávislý na programovacím jazyku. UML musí podporovat všechny rozumné programovací jazyky a vývojové procesy.
- Vytvořit základ pro porozumění modelovacího jazyka. Modelovací jazyk musí být popsán přesně a zároveň srozumitelně.
- Podpořit růst trhu s objektově orientovanými prostředky. Účastníci trhu mají sice stále možnost doplňovat své systémy o nejrůznější nástroje a funkce, ale základ modelů je jednotný a přenositelný.
- Podpořit vysokoúrovňové vývojové koncepty. Jedna s výhod objektově orientovaného návrhu je možnost opakovaného použití komponent. UML musí tento koncept definovat.

UML slouží ke:

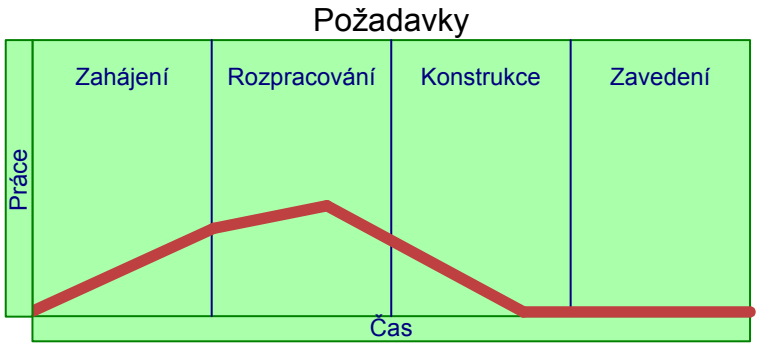
- Zakreslení. Vizualizace problému je srozumitelnější než pouhý slovní výklad. Přidají-li se k tomu jasná pravidla, co které grafické prvky

znamenají, dojdeme k rozumné hranici pro porozumění problému a vzniká vhodný dorozumívací jazyk.

- Dekompozice. Pro pochopení složitého problému je vhodné oprostit se od celku a zaměřit se na řešení dílčích konkrétních problémů.
- Tvorbě a popisu součástí systému. Popis systému slouží jako dokumentace, která poslouží uživatelům k pochopení systému a budoucím návrhářům k vývoji nových komponent v systému.

2.2 Požadavky

Požadavky lze podle standardů UML definovat jako specifikace toho, co bychom měli vytvořit. Požadavky jsou podle [1] v podstatě vyjádřením toho, co by měl systém dělat. Na proces *požadavky* budeme klást velký důraz, protože jak uvádí [1], přibližně 25% projektů končí nezdarem v důsledku problémů, které vznikly v procesu požadavků. Klíčovým zdrojem požadavků je uživatel. V procesu *požadavky* je největší množství práce věnováno fázi *zahájení* a *rozpracování*.



Obr. 2.2: Množství vykonané práce v závislosti pracovním procesu podle [1].

Po maximálním zjednodušením lze *požadavky* rozdělit podle [8] na funkční (functional requirements) a nefunkční (non-functional requirements). Jejich grafické vyjádření provedeme v procesu *analýza* pomocí diagramu případu užití (use case) s podrobným popisem aktérů (actors) a pomocí stavového diagramu.

2.2.1 Funkční a nefunkční požadavky

Funkční požadavky popisují požadovanou funkci systému. Je to formulace toho, co by měl systém dělat. Pojem ne-funkční požadavek je poněkud zavádějící, nicméně v odborných

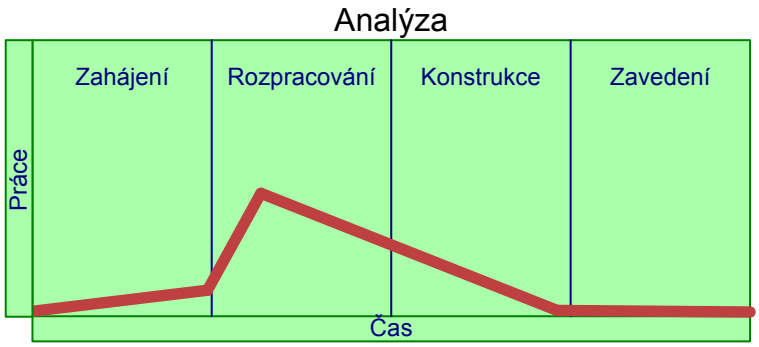
literaturách hojně používaný. Tento požadavek formuluje omezení kladená na systém nebo proces vývoje. Jinými slovy, omezující podmínka uvalená na daný systém. (Zpracováno podle [1]).

K správnému formulování požadavků bude použit jednoduchý příkaz *bude*:

<NazevSystemu>`bude`<VykonanaFunkce>

2.3 Analýza a logická úroveň modelu

Analýza spočívá v tvorbě modelů, jež zachycuje podstatné požadavky a charakteristické rysy požadovaného systému. Modelování analýzy je podle [1] strategickou aktivitou, neboť se během ní snažíme o modelování základního chování systému. Navíc [1] uvádí, že záměrem analýzy z pohledu objektově orientovaného analytika je tvorba analytického modelu. V procesu analýza je největší množství práce věnováno fázi *rozpracování* a *konstrukce*.



Obr. 2.3: Množství vykonané práce v pracovním procesu analýza podle [1].

Analytický model se zaměřuje na to, co systém musí udělat, avšak nezabývá se detaily týkající se způsobu, jakým to udělá. Tuto otázku přenecháme pracovnímu postupu *návrh* a *implementace*. Analytický model musí:

- Být v jazyce daného odvětví.
- Zachycovat problém z určité perspektivy.
- Obsahovat analytické třídy a realizovat případy užití, jež modelují problémovou doménu.
- Vyprávět příběh o požadovaném systému.

- Být užitečný pro maximální počet uživatelů a zúčastněných osob.

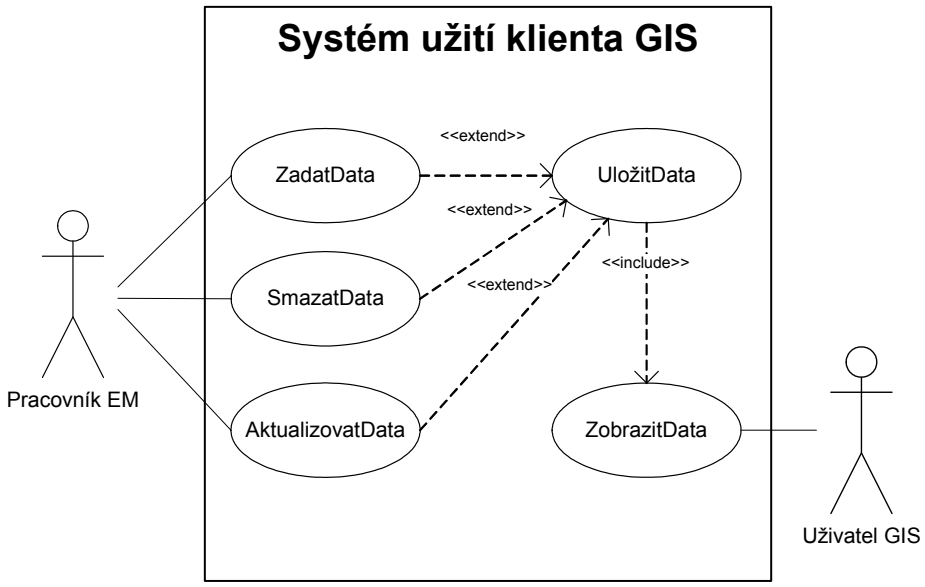
2.3.1 Případy užití a účastníci

Podle [1] případy užití zachycují funkčnost přesně tak, jak bude budoucím informačním systémem pokryta a vymezí tak jednoznačně rozsah práce. Modelování případů užití je způsob vybroušení požadavků a zároveň konstrukt, který popisuje jak se bude systém jevit potenciálním uživatelům. Modelování případů užití se skládá podle [1] z následujících aktivit:

- nalezení hranic systému,
- vyhledání aktérů,
- nalezení případů užití,
- specifikace případ užití,
- tvorba detailu případů užití,

Výstup těchto aktivit tvoří model případů užití, který obsahuje komponenty:

- *Aktéři.* Aktéři jsou vyjádřením rolí přiděleným osobám, kteří bezprostředně používají daný systém. Jak uvádí [1], aktéři jsou vůči systému externími entitami.
- *Hranice systému (system boundary).* Hranice určuje co je součástí systému a co naopak není.
- *Relace.* Relace jsou vazby mezi modelovými elementy, které mají určitý význam.



Obr. 2.4: Případ užití existujícího systému.

Popis aktérů podle obr. 2.4 - aktéři systému užití klienta GIS jsou:

- *Pracovník EM.* Pracovník evidence majetku je aktér, který fyzicky zadává data do databáze. V systému mu je přiřazena možnost přímo spouštět tři případy užití.

- *Uživatel GIS.* Uživatel GIS je každý aktér, který zobrazuje data z databáze. V systému mu je přiřazena možnost přímo spouštět případ užití.

Popis relací podle obr. 2.4 - relace použité v systému užití klienta GIS jsou:

- *Relace extend.* Tato relace je způsobem, jímž lze do existujících případů užití vložit nové chování.
- *Relace include.* Tato relace vyčleňuje společné kroky, které jsou následně zahrnuty do samostatného případu užití.

Detail případu užití - je specifikace případu užití. Výstupem této aktivity by měl být podrobnější případ užití. Pro tuto specifikaci neexistuje žádný standard UML, proto použijeme šablon podle [1], která se drží co možná nejjednodušší formy. Detail se skládá z následujících součástí:

- *Vstupní podmínky.* Jsou to kritéria, která musí být splněna ještě předtím, než je možné spustit případ užití.
- *Toky událostí.* Jsou to jednotlivé kroky případu užití.
- *Následující podmínky.* Jsou to kritéria, jež musí být splněny na konci případu užití.

Případ užití:	ZadatData
Účastníci:	Pracovník EM
Vstupní podmínky:	Vznikl nový objekt ve spravovaném majetku SVSmP?
Tok událostí:	1) Případ užití začíná při vzniku nového objektu.
	2) Pracovník EM zadá data.
	3) Systém odešle data k uložení.
Následující podmínky:	Systém uloží data.

Tab. 2.1: Detail případů užití ZadatData.

Případ užití:	SmazatData
Účastníci:	Pracovník EM
Vstupní podmínky:	Zanikl objekt spravovaného majetku SVSmP?
Tok událostí:	1) Případ užití začíná při zániku objektu.
	2) Pracovník EM vymaže data.
	3) Systém vymaže data.
Následující podmínky:	Systém uloží změnu.

Tab. 2.2: Detail případů užití SmazatData.

Případ užití:	AktualizovatData
Účastníci:	Pracovník EM
Vstupní podmínky:	Došlo ke změně hodnot objektu spravovaného majetku SVSmP?
Tok událostí:	1) Případ užití začíná při změně hodnot objektu.
	2) Pracovník EM zadá data.
	3) Systém odešle data k uložení.
Následující podmínky:	Systém uloží data.

Tab. 2.3: Detail případů užití AktualizovatData.

Případ užití:	UložitData
Účastníci:	Systém
Vstupní podmínky:	Došlo k případu užití ZadatData, AktualizovatData nebo SmazatData?
Tok událostí:	1) Případ užití začíná při zadání, aktualizaci nebo smazáním dat. 2) Systém data uloží.
Následující podmínky:	Systém odešle data k zobrazení.

Tab. 2.4: Detail případů užití UložitData.

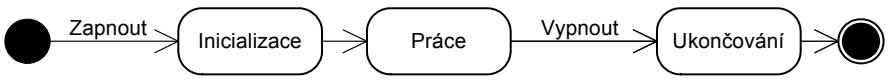
Případ užití:	ZobrazitData
Účastníci:	Uživatel GIS
Vstupní podmínky:	Jsou požadovaná data uložena v databázi?
Tok událostí:	1) Případ užití začíná při zobrazení požadovaných dat. 2) Systém zobrazí data.
Následující podmínky:	-

Tab. 2.5: Detail případů užití ZobrazitData.

2.3.2 Stavový diagram

Systém je převážně tvořen grafickým uživatelským rozhraním (GUI). Stavový diagram podle [15] reprezentuje všechny stavy, do nichž se může aktér případu užití v GUI dostat, a také podmínky přechodů mezi jednotlivými stavy. Systémové změny v tomto diagramu jsou popsány, tak že požadavky na systém mění v čase svůj stav.

K označení stavu slouží obdélník se zaoblenými rohy. Šipka představuje přechod mezi stavy a směřuje do stavu, do nějž aktér přechází. K označení počátku slouží plný kroužek, k označení konce kroužek s bílým okrajem.

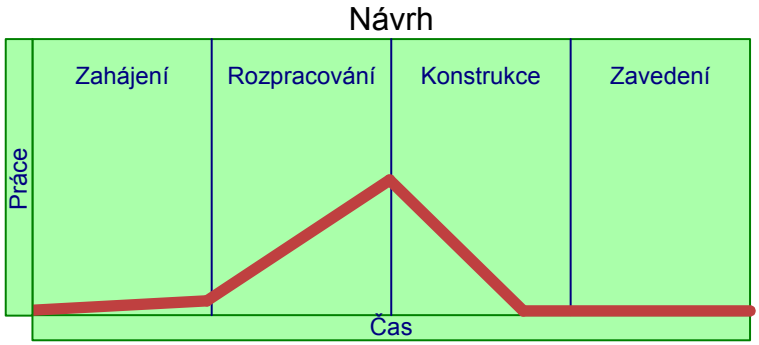


Obr. 2.5: Stavy a přechody grafického uživatelského rozhraní pro všechny aktéry.

Výsledkem akcí provedených ve stavu *inicializace* je, že GUI přejde do stavu *práce*. Rozhodne-li se uživatel svůj počítač vypnout, spustí událost, která způsobí přechod GUI ze stavu *práce* do stavu *ukončování* a počítač se vypne.

2.4 Návrh a fyzická úroveň modelu

Smyslem návrhu je podle [1] přesná specifikace způsobu, jak implementovat funkce zachycené v analýze. V procesu *návrh* je nejvíce množství práce věnováno ve fázi *rozpracování* a *konstrukce*.



Obr. 2.6: Množství vykonané práce v pracovním procesu návrh podle [1].

Návrh spočívá ve sloučení technických řešení za účelem vytvoření návrhového modelu, který skutečně lze implementovat. Návrhový model lze považovat za rozpracovanou formu analytického modelu, do něhož jsou přidány detaily a specifická technická řešení. Návrhový model obsahuje stejné typy předmětů jako analytický model, ale veškeré artefakty jsou již plně vytvořeny a musí nyní obsahovat rovněž implementační detaily.

Návrhový model obsahuje:

- jeden návrhový systém,
- návrhový podsystém,
- návrh realizace případů užití,
- rozhraní,
- návrhové třídy,
- první verzi diagramu nasazení.

Návrh je dalším krokem při návrhu datového modelu na fyzické úrovni. Fyzická datová úroveň již zohledňuje zvolenou databázi, v našem případě relační MS SQL Server 2000.

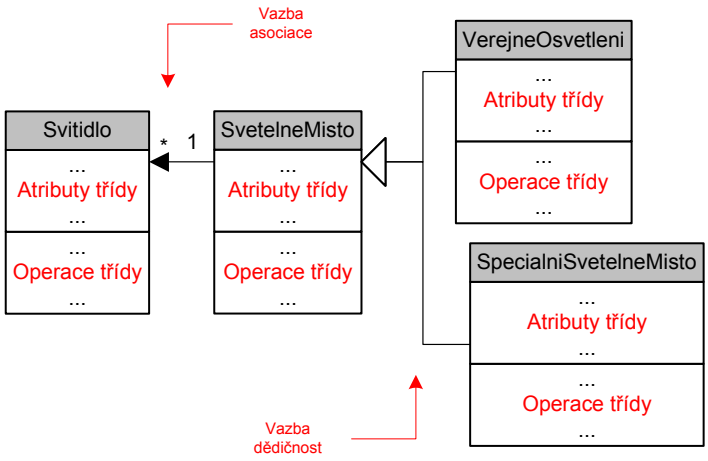
2.4.1 Diagram objektových tříd

Diagram objektových tříd je strukturní diagram. Třídy a objekty jsou základními stavebními bloky všech objektově orientovaných systémů. Objekt kombinuje data a funkce do jedné soudržné jednotky, která se vyznačuje:

- *Identitou*, jedinečnou identifikací objektu, která jej odlišuje od všech ostatních objektů.
- *Stavem*, přesným určením pomocí hodnot atributů.
- *Chováním*, specifikací toho, co pro nás objekt může udělat. Této specifikaci říkáme v procesu *analýza* množina operací, v procesu *návrh* množina metod.

Každý objekt je instancí přesně jedné třídy. Třídy popisují charakteristické rysy určité množiny objektů, které spojuje určitá relace.

Pro návrh objektových systémů se předpokládá, že je třeba navrhnout model tříd objektů. Třída objektů je definována svými atributy, operacemi (metodami), relacemi a chováním (více viz. např. [22]).

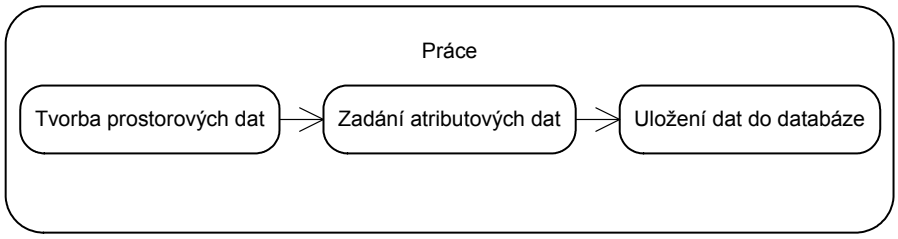


Obr. 2.7: Ukázka diagramu objektových tříd systému veřejného osvětlení.

2.4.2 Podstavy stavového diagramu

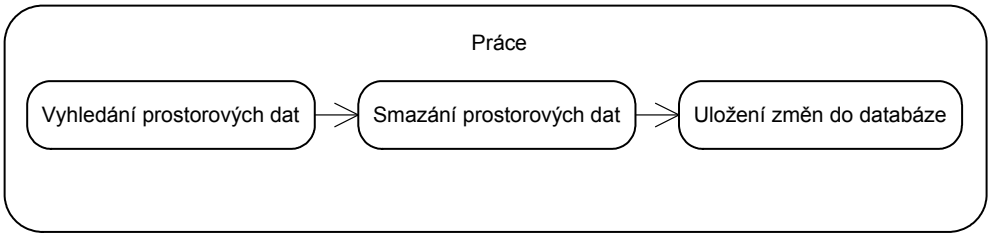
Stav objektu ve stavovém diagramu se může skládat z více podstavů, jako je to ve stavu *práce*. Podstavy stavového diagramu jsou procesní diagramy a mohou být sekvenční nebo souběžné. V našem případě se jedná o podstavy sekvenční.

K označení stavu slouží obdobně jako u stavového diagramu obdélník se zaoblenými rohy. Podstavy se značí rovněž obdélníkem se zaoblenými rohy uvnitř stavového obdélníku. Šipka představuje přechod mezi podstavy a směřuje do podstavu, do kterého aktér přechází.



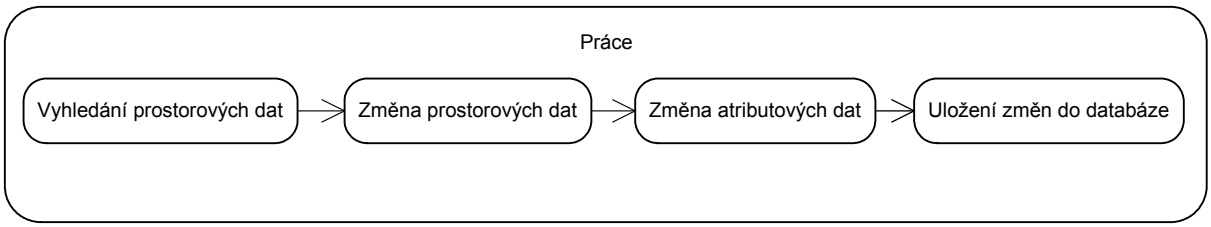
Obr. 2.8: Podstavy ve stavu práce pro aktéra pracovník EM v ZadatData.

Hlavní činnost *pracovníka* evidence majetku je přímo zadávat geodata. Při této činnosti *pracovník* postupuje tak, že nejprve zakreslí prostorová data. K prostorovým datům připíše atributová data. Poté takto vytvořená geodata uloží do databáze.



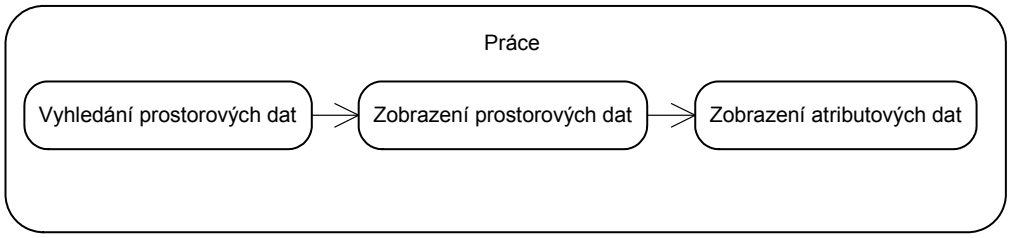
Obr. 2.9: Podstavy ve stavu práce pro aktéra pracovník EM v SmazatData.

Systém umožňuje *pracovníkovi* evidence majetku geodata mazat. *Pracovník* musí nejprve vyhledat prostorová data, která chce smazat. Data smaže, čímž dojde i k smazání atributových dat, které byly provázány s konkrétními prostorovými daty. Poté všechny změny uloží do databáze.



Obr. 2.10: Podstavy ve stavu práce pro aktéra pracovník EM v AktualizovatData.

Další činností je aktualizace geodat. *Pracovník* evidence majetku vyhledá prostorová data, která chce aktualizovat. Tato geodata změní v pořadí, nejprve prostorová a poté atributová. Dále *pracovník* uloží změny do databáze.



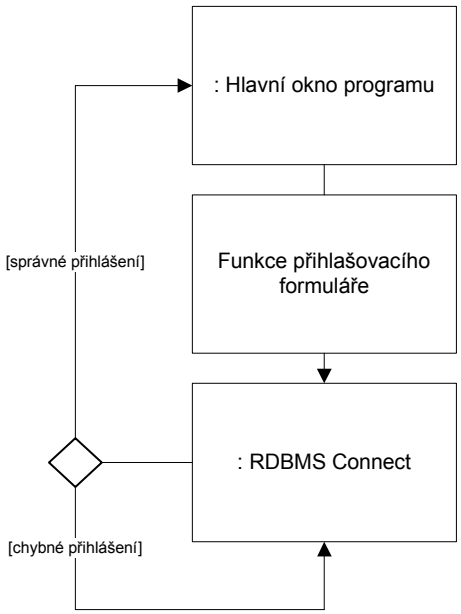
Obr. 2.11: Podstavy ve stavu práce pro aktéra uživatel GIS v ZobrazitData.

Hlavní činností uživatele GIS je zobrazovat geodata. Při této činnosti uživatel vyhledá prostorová data, která zobrazí. K zobrazeným datům si zobrazí atributová data.

2.4.3 Diagram grafického uživatelského rozhraní

Grafické uživatelské rozhraní (GUI) je prostředí v jaké uživatel zadává data. Uživatelské rozhraní se skládá navíc z položek nabídek a sestav. GUI lze tedy definovat jako všechno co uživatel vidí nebo s čím pracuje. Diagram GUI není standardem jazyka UML. Nicméně v naší analytické práci je třeba se o něm zmínit, protože systém, který je popsán a později i bude vyvíjen, je převážně tvořen GUI. Diagram GUI si rozdělíme na diagram toku a prototyp GUI. Diagram toku nám umožní definovat postup v programu. Prototyp GUI nám umožní definovat obsah toho, co uživatel vidí na obrazovce.

Diagramy toku se nejčastěji používají k určení, jak bude uživatel se systémem pracovat v konkrétním stavu. V našem diagramu toku ukážeme, jak aktéři případu užití postupují ve stavu *inicializace*, protože tento stav je pro všechny aktéry stejný.

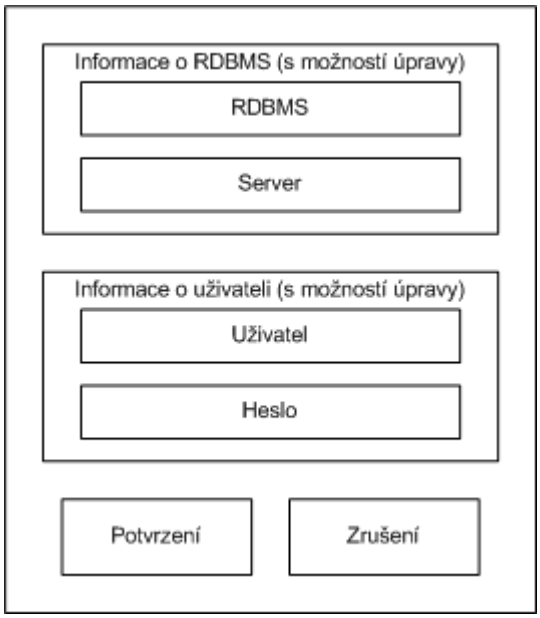


Obr. 2.12: Diagram toku popisující GUI pro příklad inicializace.

Popis diagramu toku:

- Aktér vybere v hlavním okně programu možnost přihlásit se.
- Zobrazí se přihlašovací formulář RDBMS Connect, ve kterém aktér zadá:
  - název databáze,
  - uživatelské jméno,
  - uživatelské heslo,
  - název serveru.
- Jestliže přihlášení proběhne v pořádku, potom se přihlašovací formulář zavře.
- Jinak je nutno zadat správné údaje.

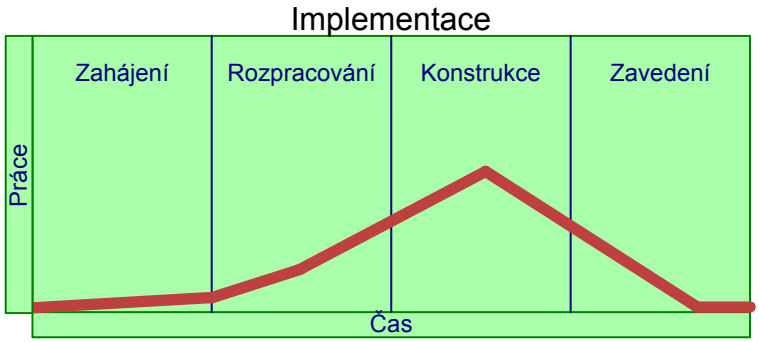
Prototyp GUI slouží k definování obsahu, který bude zobrazován v určitém formuláři.



Obr. 2.13: Prototyp GUI pro příklad inicializace.

2.5 Implementace

Implementace je hlavním pracovním postupem ve fázi *konstrukce*. Jak uvádí [1], spočívá v převodu navrhovaného modelu do spustitelného prostředí. Z našeho pohledu je smyslem implementace tvorba implementačního modelu, který určí jak navrhované třídy budou pracovat v konkrétním programu.

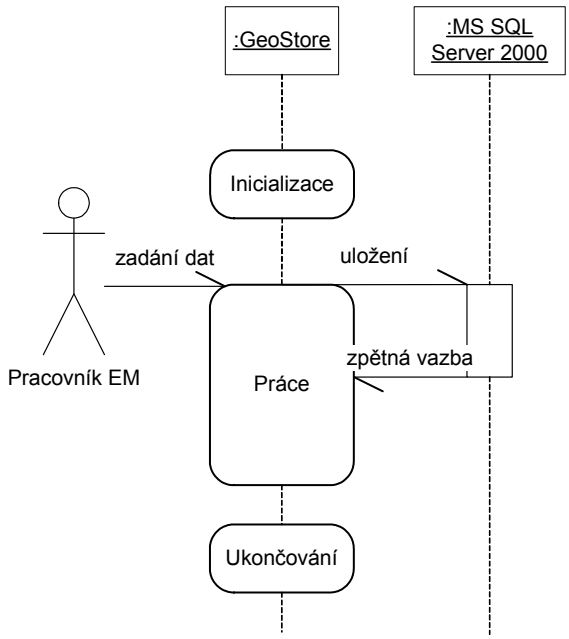


Obr. 2.14: Množství vykonané práce v závislosti pracovním procesu podle [1].

2.5.1 Sekvenční diagram

„Sekvenční diagram jazyka UML doplňuje přehled interakcí mezi objekty o rozměr času“ [15]. Sekvenční diagram je procesní diagram, je dvourozměrný a popisuje, jak spolu objekty v čase komunikují. Sekvenční diagram popíše interakci GUI s dalšími objekty, kde klade důraz především na pořadí jednotlivých událostí.

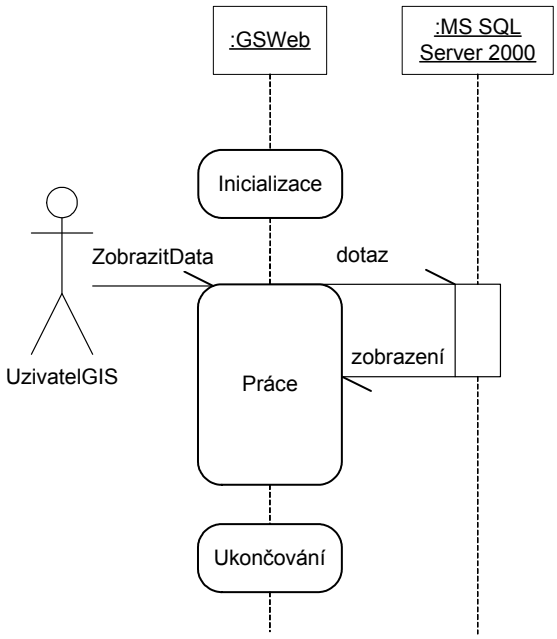
V sekvenčním diagramu se ubíhající čas zobrazuje vertikálně směrem dolů. Objekty jsou zakresleny vedle sebe zleva doprava a znázorňují se jako obdélníky na horní hraně diagramu. Z každého objektu vede svislá čára reprezentující život objektu v průběhu chování v daném případu užití. Každá zpráva mezi objekty je znázorněna šipkou mezi čárami života dvou zúčastněných objektů. Zpráva znamená, že jeden objekt vyvolá operaci druhého objektu. Pořadí, ve kterém jsou zprávy zasílány je dáno směrem shora dolů.



Obr. 2.15: Sekvenční diagram zachycující všechny případy užití aktéra pracovník EM.

Popis sekvencí:

- *Pracovník EM* zadá nová data, smaže původní data nebo aktualizuje původní data v programu GeoStore. Tyto operace jsou souhrnně uskupené do zprávy *zadání dat*.
- GeoStore předá MS SQL Serveru 2000 zprávu o uložení zadaných dat ve zprávě *uložení*.
- MS SQL Server 2000 o tom zpětně informuje GeoStore zprávou *zpětná vazba*.



Obr. 2.16: Sekvenční diagram zachycující případ užití aktéra uživatel GIS.

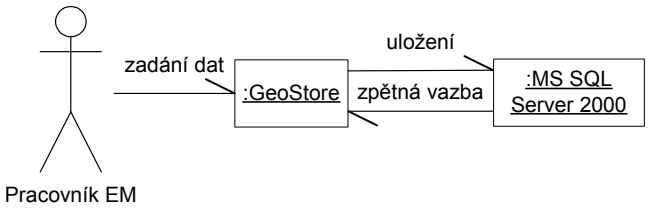
Popis sekvencí:

- *Uživatel GIS* zobrazí data v programu GS Web.
- Programu GS Web se dotáže MS SQL Server 2000 na data zprávou *dotaz*.
- MS SQL Serveru 2000 odešle data programu GS Web zprávou *zobrazení*.
- GS Web data zobrazí.

2.5.2 Diagram spolupráce

Jako alternativní techniku k sekvenčnímu diagramu můžeme použít diagram spolupráce. Diagram spolupráce je stejně jako sekvenční diagram procesním diagramem a zobrazuje interakci mezi objekty. Diagram zobrazuje jednotlivé objekty spolu se zprávami, které se zasílají ostatním objektům. Od diagramu sekvencí se liší tím, že zdůrazňuje spíše kontext a uspořádání spolupracujících objektů. Forma prostorového vzhledu diagramu dovoluje lépe znázornit spojení objektů v rámci jejich spolupráce. (Zpracováno podle [15]).

Na diagramu spolupráce jsou objekty znázorněny obdélníky, vazby mezi nimi pomocí spojnic a šipky indukují zprávy zasílané mezi objekty v rámci případu užití.

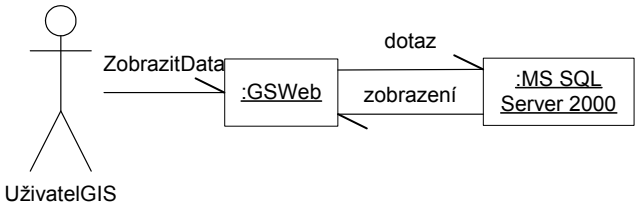


Obr. 2.17: Diagram spolupráce zachycující všechny případy užití aktéra pracovník EM.

Popis spolupráce:

- *Pracovník EM* zadá nová data, smaže původní data nebo aktualizuje původní data v programu GeoStore. Tyto operace jsou souhrnně uskupené do zprávy *zadání dat*.
- GeoStore předá MS SQL Serveru 2000 zprávu o uložení zadaných dat ve zprávě *uložení*.

MS SQL Server 2000 o tom zpětně informuje GeoStore zprávou *zpětná vazba*.



Obr. 2.18: Diagram spolupráce pro GUI.

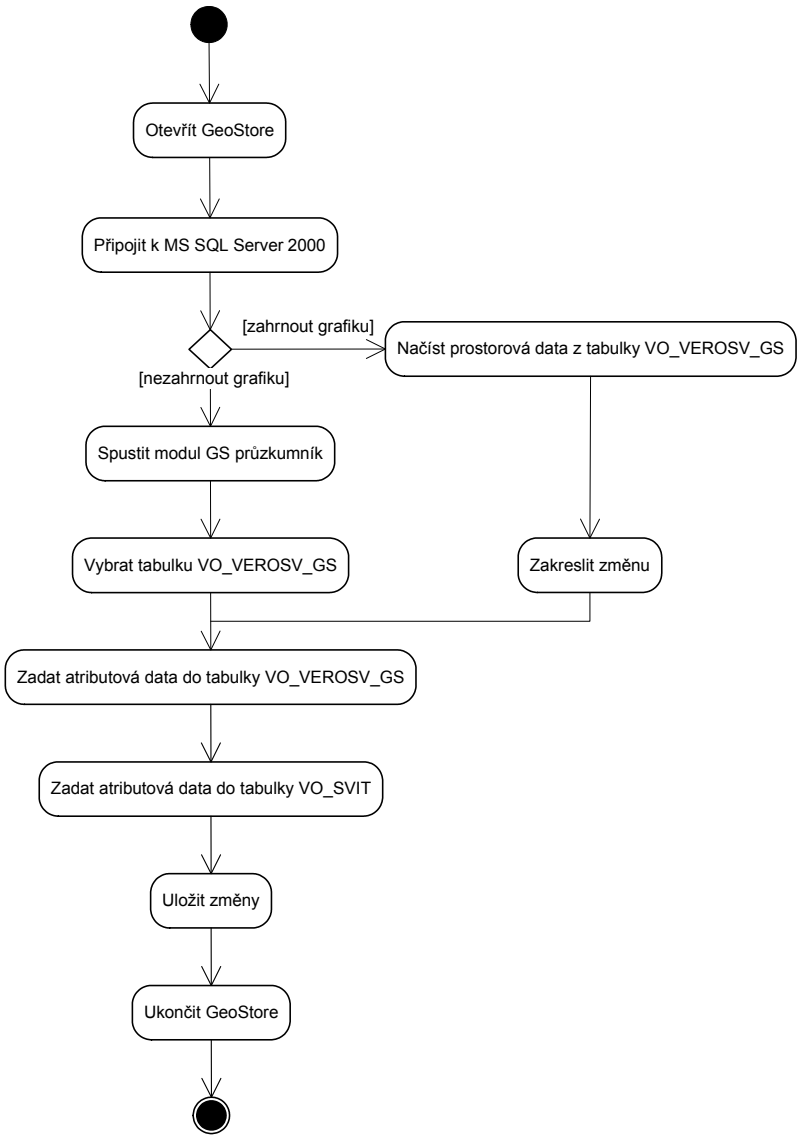
Popis spolupráce:

- *Uživatel GIS* zobrazí data v programu GS Web.
- Programu GS Web se dotáže MS SQL Server 2000 na data zprávou *dotaz*.
- MS SQL Serveru 2000 odešle data programu GS Web zprávou *zobrazení*.
- GS Web data zobrazí.

2.5.3 Diagram činností

Diagram činností je procesní diagram a zachycuje kroky, rozhodování a větvení sekvencí. Diagram činností je podle [15] prostředek, který umožňuje popsat jednotlivé kroky procesu. Diagram vzniká rozšířením stavového diagramu.

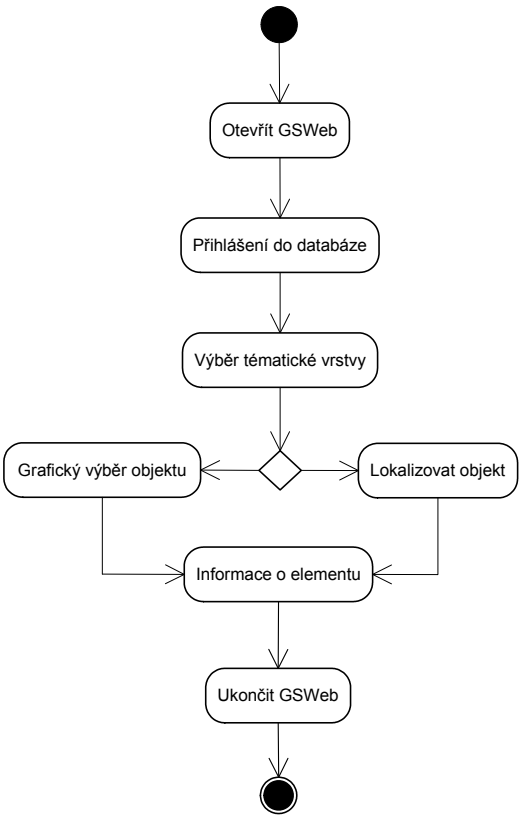
Každá činnost je reprezentována obdélníkem se zaoblenými rohy. Vždy po dokončení příslušné činnosti následuje automaticky přechod na další činnost. Tento přechod je v diagramu činností naznačen šipkou. Počátek diagramu činností je označený plným kroužkem a konec označený kroužkem s bílým okrajem. Místa, kde je třeba provést rozhodnutí lze označit, takže se tok řízení nejprve přesune z aktivity do malého kosočtverce a teprve odtud se rozvětví do dvou možných cest. Přitom se obě cesty navzájem vylučují. Cesty jsou doplněny o podmínku uvedenou v hranatých závorkách.



Obr. 2.19: Diagram činností popisující práci pracovníka EM.

Popis činnosti *pracovníka EM*:

- Otevře program GeoStore.
- Provede správné přihlášení.
- Jestliže bude kreslit prostorový prvek, potom načte prostorová data z tabulky a zakreslí změnu.
- Jinak spustí modul GS průzkumník, ve kterém vybere tabulku.
- Zadá atributová data do tabulky.
- Zadá atributová data do podřadné tabulky.
- Uloží změny.
- Ukončí program GeoStore.



Obr. 2.20: Diagram činností popisující práci uživatele GIS.

Popis činnosti *uživatele GIS*:

- Otevře program GS Web.
- Provede správné přihlášení.

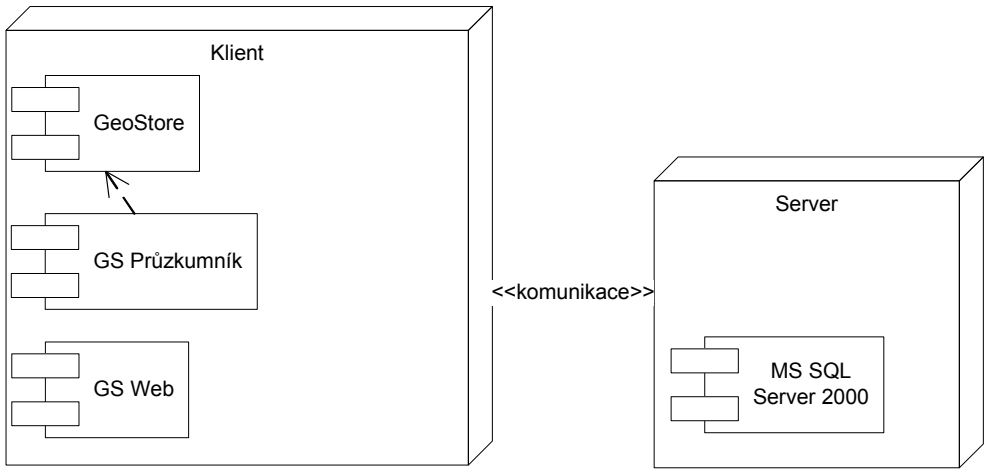
- Vybere tématickou vrstvu.
- Jestliže zná jednoznačný identifikátor objektu, potom provede lokalizaci objektu podle tohoto identifikátoru.
- Jinak provede výběr pomocí grafiky.
- Získá atributové informace o objektu.
- Ukončí program GS Web.

2.5.4 Diagram komponent a jejich nasazení

Diagram komponent je strukturní diagram, zobrazuje komponenty, rozhraní a jejich vztahy. Zachycuje skutečné objekty - softwarové komponenty. Softwarové komponenty jsou uloženy v paměti počítače. Komponenty a jejich vztahy se modelují s následujících důvodů:

- Aby mohla být zákazníkovi předvedena hotová struktura systému.
- Aby vývojáři věděli co a s čím budou dělat.
- Aby osoby pověřené sepsáním dokumentace věděly o čem psát.
- Aby si analytici usnadnili opakované použití komponenty.

Hlavně poslední bod skýtá jednu z nejužitečnějších vlastností komponent. Tato vlastnost je důvodem současné vize, podle které budoucnost vývoje softwaru patří právě komponentovému vývoji. *„V dnešní době záleží na rychlosti, s jakou dokážete připravovaný systém uvést do chodu“* [15]. Máme-li vyvinutou komponentu, můžeme ji uplatnit i v dalších projektech.



Obr. 2.21: Diagram komponent a jejich nasazení.

3. Vývoj systému užití klientů GIS.

Tato část práce se bude krok po kroku zabývat postupem analytických prací při vývoji systému. Vývoj systému bude ukázán na příkladu evidence veřejného osvětlení (VO). Při analytických postupech budeme využívat UML techniky a poznatků ze SWOT analýzy (kap.1.6.3). Pro optimalizaci využívání geografických informací bude v zásadě vytvořen jednoduchý klient GIS pro zadávání popisných informací k již existujícímu grafickému prvku. Tento klient bude přidělen novému aktérovi *technik*. GUI bude co možná nejjednodušší, aby bylo srozumitelné pro *technika*, který doposud předával informace aktérovi *pracovník EM*. Aktér *pracovník EM* je následně zadával do databáze. Pro optimalizaci budou tedy aktérovi *technik* přidělena práva aktualizovat údaje z centrální databáze.

Celý proces přidání aktéra *technik*, který může aktualizovat data, lze obdobně aplikovat i pro jiné obory správy dat na SVSmP, tím se zvýší množství uchovávaných dat a zároveň se neomezí jejich kvalita. V této části bude opomíjen aktér *uživatel GIS*, jehož práce je v dosavadním i optimalizovaném systému stejná a pro pochopení systému a jeho další vývoj není důležitá.

3.1 Vývoj systému pomocí UML

V kapitole 2.2 byly *požadavky* rozděleny na funkční a nefunkční. Tohoto rozdělení se budeme držet i v požadavcích při vývoji systému.

3.1.1 Funkční požadavky

Při vývoji systému, která se opírá o standard UML budeme vycházet z teorie a popisu doposud existujícího systému z kap. 2. Zahájíme tedy práci na vylepšování systému a vývoj popíšeme pomocí UML. Obecně, požadavky jsou tím, čím se setkáme hned na samém začátku. První funkční požadaveky se opírají o přání *techniků* VO. Dalšími požadavky jsou pak všeobecné požadavky na softwarového klienta pro práci s geodaty:

- Do databáze se budou ukládat informace:
  - typ VO,
  - číslo VO,
  - celkový příkon VO,

- popis umístění VO,
- majitel VO,
- napojení na rozvaděč,
- číslo zakázky,
- materiál podpěry VO,
- výška podpěry VO,
- rok osazení podpěry VO,
- materiál patice podpěry VO,
- odstín nátěru podpěry VO,
- rok nátěru podpěry VO,
- počet ramen výložníku VO,
- materiál ramen výložníku VO,
- velikost ramene výložníku VO,
- název svítidla,
- typ svítidla,
- rok osazení nebo výměny svítidla,
- příkon svítidla,
- zdroj svítidla,
- rok osazení nebo výměny zdroje svítidla.

- Klient GIS bude jednoduchý na práci s aktualizovanými daty.

3.1.2 Nefunkční požadavky

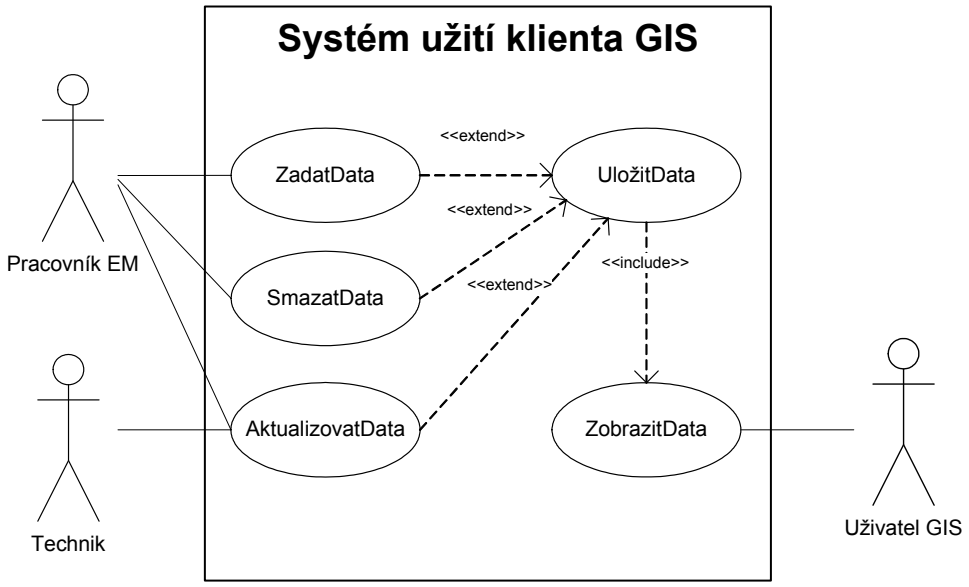
Nefunkční požadavky budou tvořit omezení kladené na systém:

- Klient GIS bude ukládat atributová data do centrální databáze.
- Aktualizace objektu bude probíhat pomocí HTML formuláře.
- Založení prostorového prvku nebude úlohou nově vzniklého klienta GIS.
- Založení nového prostorového prvku bude v programu GeoStore.

3.1.3 Diagram případu užití

Poté co byly nadefinovány *požadavky*, lze vytvořit první diagram případu užití nově zkomponovaného systému. Diagram obsahuje nově aktéra *technik*, o kterém bylo krátce

zmíněno v úvodu kapitoly. Tento aktér je pro optimalizovaný systém zásadní a na něm je postaven vývoj systému. Proto se podívejme na případ užití.



Obr. 3.1: Případ užití optimalizovaného systému.

Případ užití je téměř totožný s případem užití z kap. 2.2. Aktér *pracovník EM* může spouštět stejné případy užití jako v dosavadním systému. Přidán je aktér *technik*, kterému je povoleno spouštět přímo případ užití *AktualizovatData.*

Nyní lze pokračovat v analytickém postupu vytvořením scénáře případu užití pro nového aktéra *technik*.

Detail případu užití:

Případ užití:	AktualizovatData
Účastníci:	Technik
Vstupní podmínky:	Došlo ke změně uložených dat?
Tok událostí:	1) Případ užití začíná při změně uložených dat.
	2) Technik zadá data.
	3) Systém odešle data k uložení.
Následující podmínky:	Systém uloží data.

Tab. 3.1: Detail případů užití ZobrazitData.

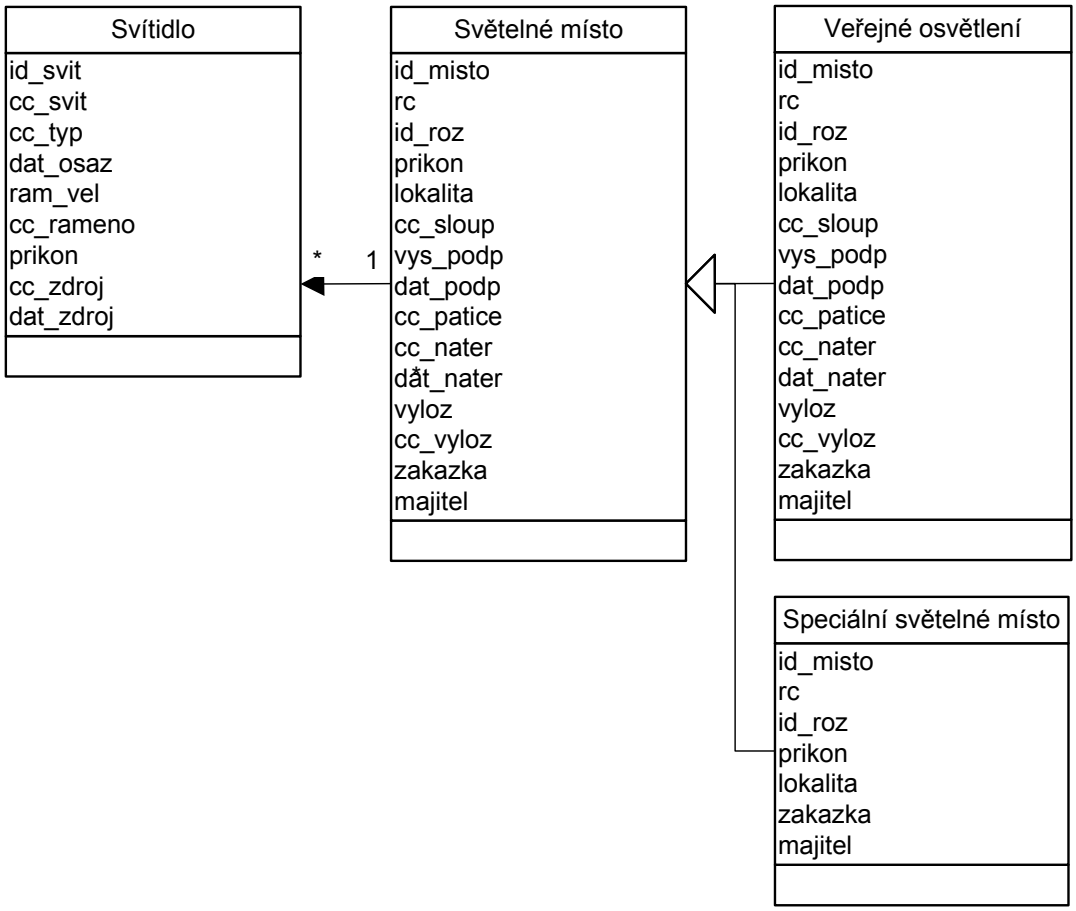
3.1.4 Stavový diagram systému

Stavový diagram popisovaný v kap. 2.3.2 je velmi obecný tak aby splňoval podmínky pracovního procesu *analýza*. Tento diagram (obr. 2.5) platí i pro nového aktéra *technik*, který se podle stavového diagramu může taktéž dostat do tří stavů ve svém případu užití:

- *inicializace*,
- *práce*,
- *ukončování*.

3.1.5 Diagram objektových tříd

Osobně upřednostňuji kompletní dokončení tvorby případu užití a poté začít s tvorbou diagramů případů užití, proto také byly tyto diagramy v teoretické části začleněny do pracovního procesu *návrh*. Nicméně mnoho analytiků tvoří diagram objektových tříd zároveň s případem užití. Náš diagram objektových tříd byl vytvořen na základě jasných požadavků techniků VO. Tyto požadavky byly převedeny na atributy uchovávající hodnoty, které chtěli technici k VO uchovávat.



Obr. 3.2: Ukázka diagramu objektových tříd systému VO.

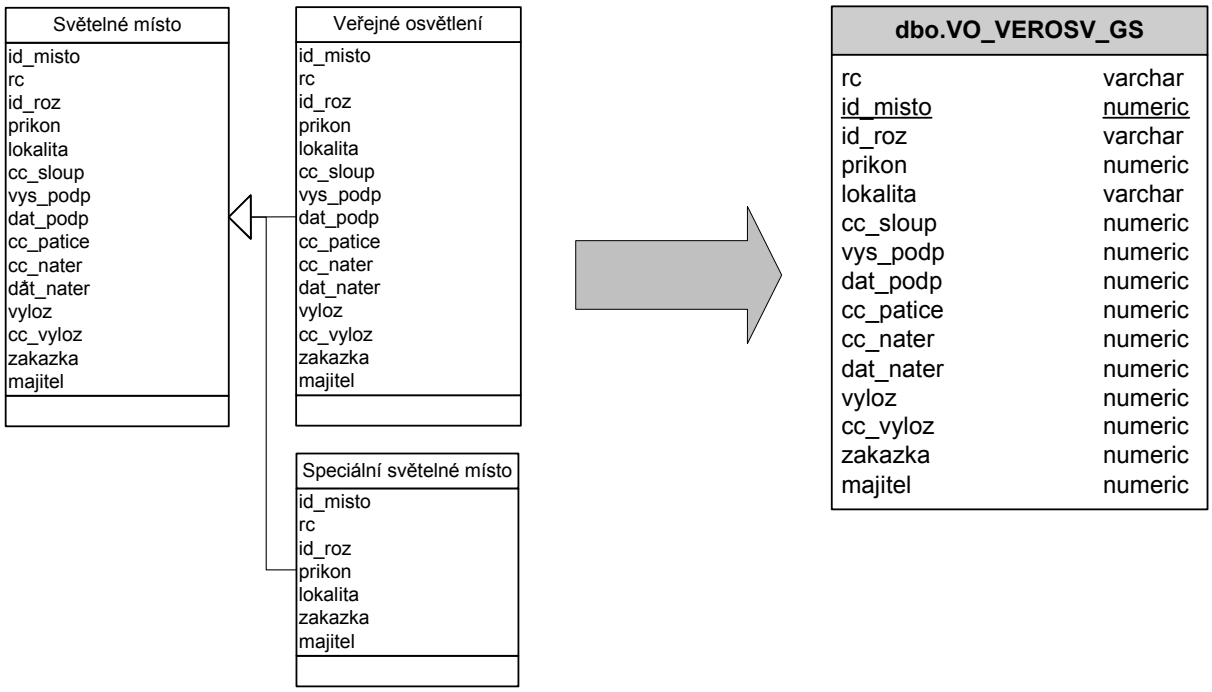
Diagram je velmi jednoduchý a obsahuje čtyři objektové třídy. Hlavní třída *Světelné místo* může mít několik tříd *Svítidlo*. Zároveň se hlavní třída *Světelné místo* skládá z tříd *Veřejné osvětlení* a *Speciální světelné místo*.

Systém evidence VO je popisován v objektově-orientovaném jazyce a především na diagramech v syntaxi UML. Nicméně datový sklad pracuje na relační bázi, které je třeba se přizpůsobit, protože organizace má všechna data uložena právě v tomto relačním datovém skladu. Pro vytvoření návrhu fyzického datového modelu podle požadavků a analýz bude potřeba objektové třídy z této kapitoly mapovat na relační tabulky. Datový model bude vytvořen tak, aby mohl rozumně získávat data z tabulek relační databáze.

3.1.6 Mapování tříd na tabulky

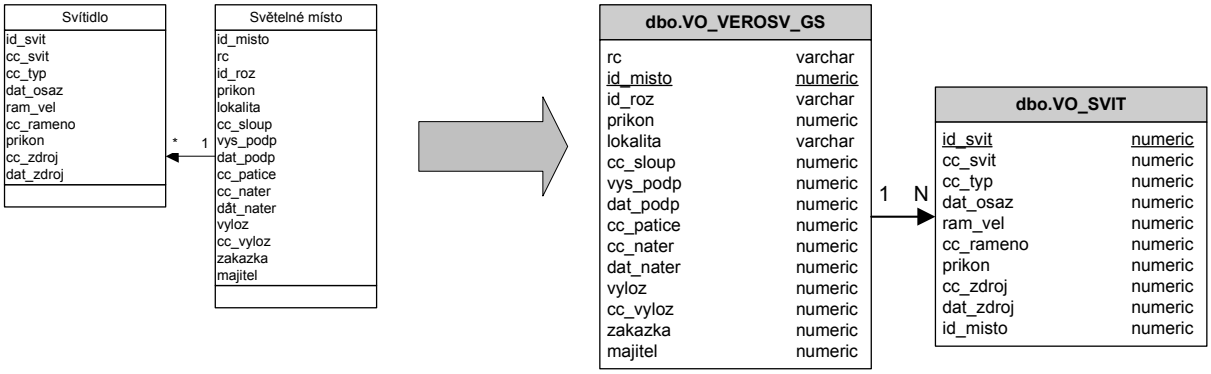
Kapitola se týká datového modelování. Při mapování se atributy třídy stanou sloupci tabulky. Pomocí mapování dědičnosti se zahrnou podtřídy do nadtřídy. Tento způsob je vhodný v případě menšího počtu podtříd. Na obr. 3.3 je vidět jak vznikne ze dvou

objektových tříd relační tabulka, kde jednomu atributu odpovídá jeden sloupec tabulky. Relační tabulka dostane název VO_VEROSV_GS.



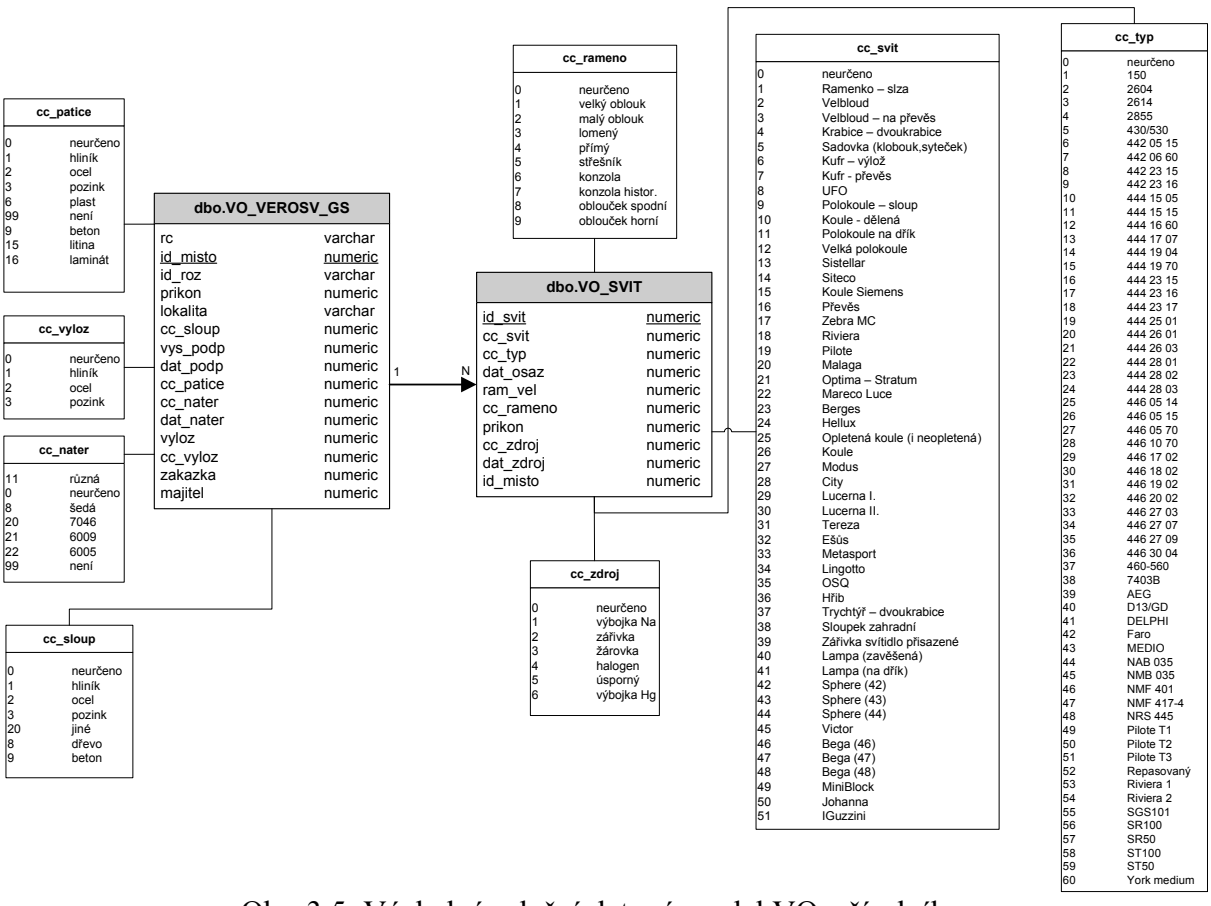
Obr. 3.3: Mapování dědičnosti zahrnutím podtříd do nadtříd.

Pomocí mapování asociace 1:N bude určen cizí klíč v podřazené tabulce, který odkazuje na příslušný řádek v nadřazené tabulce. Na obr. 3.4 je vidět převod dvou objektových tříd s vazbou 1:N na dvě relační tabulky s vazbou 1:N. Cizím klíčem se v podřazené tabulce stane atribut *id_misto* a tím bude zachována vazba 1:N i v relačním datovém modelu. Podřízená relační tabulka dostane název VO_SVIT.



Obr. 3.4: Mapování asociací 1:N.

Na obr. 3.5 je vidět výsledný relační datový model i s hodnotami číselníků. Číselníky byly vytvořeny na základě požadavků *techniků* VO.



Obr. 3.5: Výsledný relační datový model VO s číselníky.

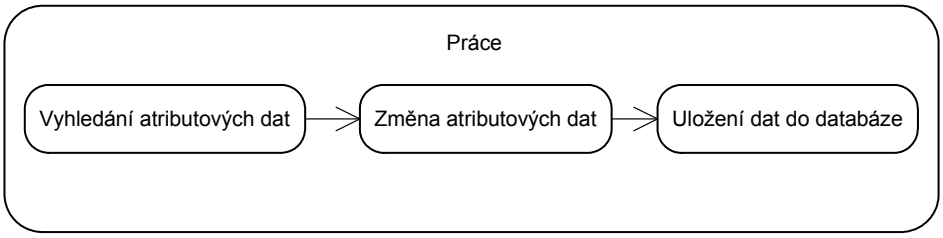
Byl vytvořen datový model systému a zároveň jeho dokumentace. Výsledný relační model popisuje uchování dat v databázi prostřednictvím tabulek. Pomocí relačního modelu bude napsán kód v jazyce SQL (Structured Query Language), který bude implementován

do databáze. Tato část práce zároveň poskytuje dostatečnou dokumentaci pro případný zásah do budovaného systému.

3.1.7 Podstavy stavového diagramu

Podstavy ve stavovém diagramu pro aktéra *pracovník EM* ve stavu *práce* byly pro různé případy užití. Tyto podstavy zůstávají stejné.

Ve vývoji systému platí pro aktéra *technik* nové podstavy stavového diagramu pro jeho případ užití *AktualizovatData*, které jsou opět sekvenční. *Technik* oproti *pracovníkovi EM* neaktualizuje prostorová data.

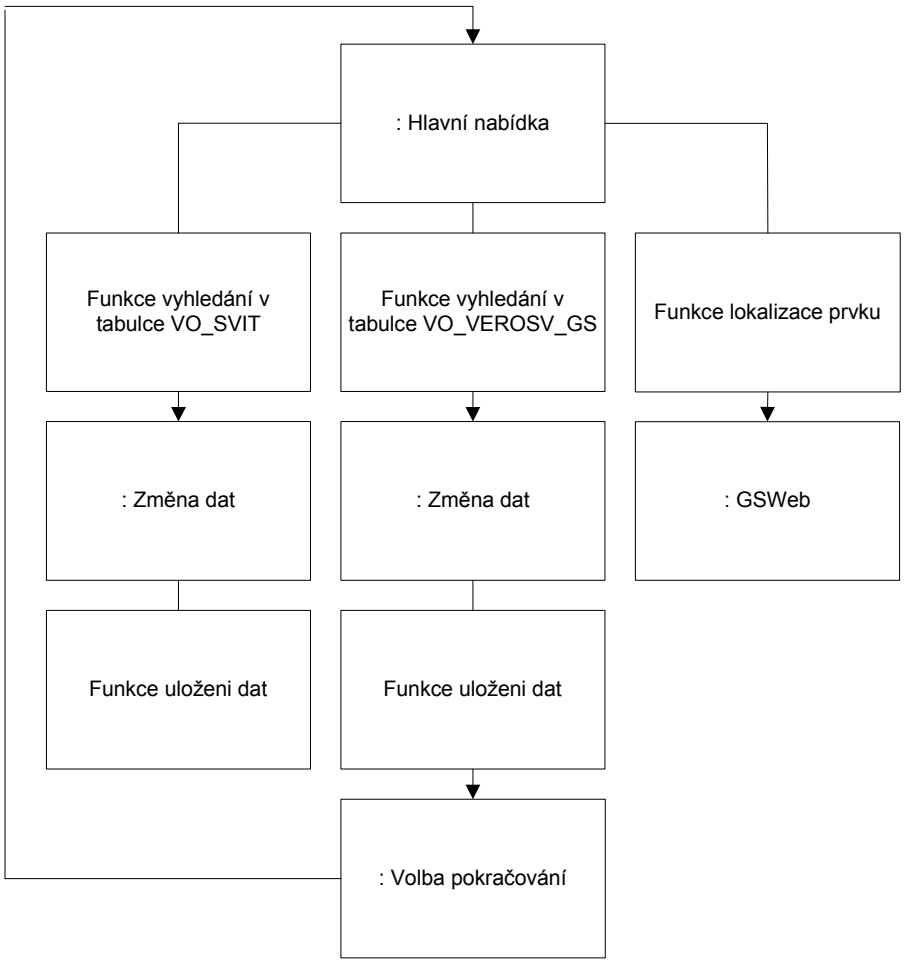


Obr. 3.6: Podstavy ve stavu práce pro aktéra technik v případu užití AktualizovatData.

Technik přímo spouští případ užití *AktualizovatData*, kde se po *inicializaci* dostává do stavu p*ráce.* Ve stavu *práce technik* postupuje tak, že postupně provede tři sekvenční podstavy *vyhledání atributových dat*, dále *změna atributových dat* a nakonec *uložení dat do databáze.*

3.1.8 Diagram grafického uživatelského rozhraní

Diagram toku a jeho prototyp zůstává pro případ přihlášení stejný (viz. kap.2.4.3) pro všechny aktéry. V případě vývoje systému bude vytvořeno nový klient GIS, které bude používat *technik*. Diagram toku nám umožní definovat, jak aktér *technik* bude pracovat s konkrétním případem užití *AktualizovatData*.



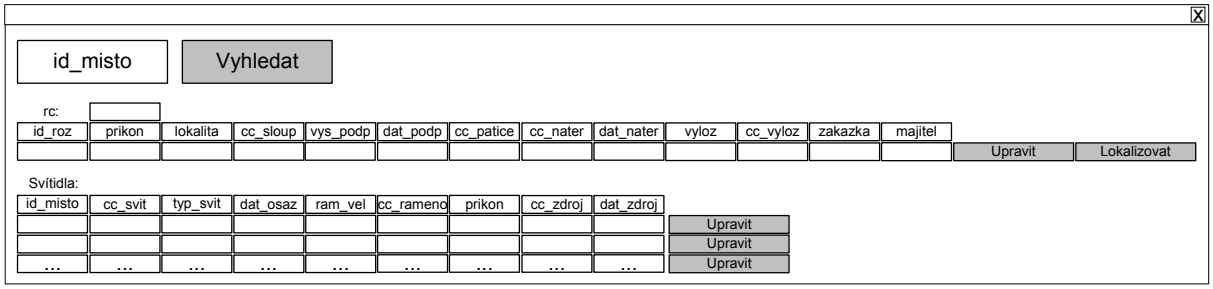
Obr. 3.7: Diagram toku popisující práci aktéra technik pro případ užití AktualizovatData.

Popis diagramu toku:

- *Technik* zadá v hlavní nabídce identifikátor objektu a zadá příkaz vyhledat.
- Jestliže *technik* vybere funkci *Změna dat* v tabulce osvětlení, potom se zobrazí atributová data a *technik* je může aktualizovat.
- Jestliže *technik* vybere funkci *Lokalizovat prvek*, potom se otevře program GS Web s prostorovým vyjádřením objektu.
- Jestliže *technik* vybere funkci *Změna dat* v tabulce svítidel, potom se zobrazí atributová tabulka svítidel a *technik* je může aktualizovat.

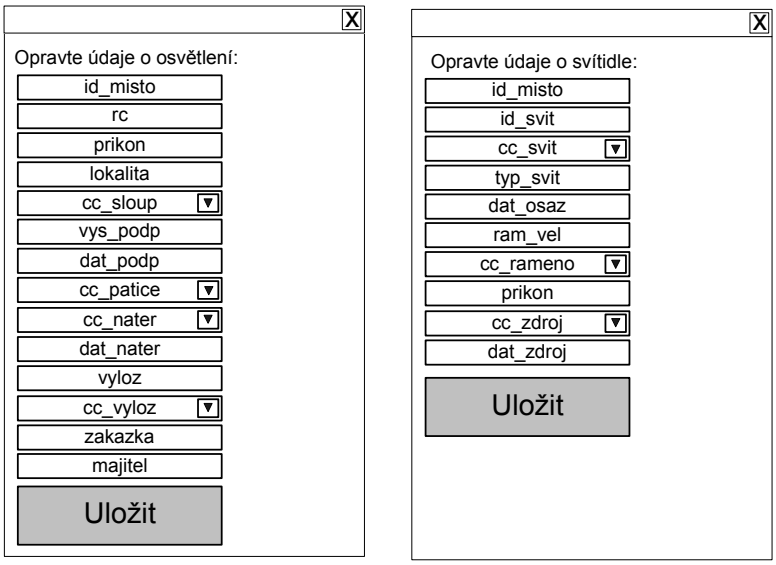
Prototyp GUI slouží k definování obsahu, který bude zobrazován ve formuláři pro aktéra *technik*. Podle [15] jsou popsány základní zásady návrhu GUI na základě jejich zkušeností:

- Je nutné pochopit, co má uživatel dělat.
- Uživatel musí mít ovládání plně pod kontrolou.
- Uživatel musí mít více možností pro dokončení každé činnost.
- Nejdůležitější informace musí být umístěny v levém horním rohu obrazovky.
- Prvky, které spolu souvisí musí být blízko sebe.
- Pro vysvětlování pojmů se používá činný slovesný rod.
- Příliš mnoho barev uživatele rozptyluje.
- Uživatel mnohdy vztah mezi barvou a významem určitého prvku nevidí.
- Nepoužívá se kurzíva a ozdobná písma.
- Tlačítka a výběrové seznamy musí být stejně velká.
- Tlačítka, výběrové seznamy a datové položky musí být zarovnané podle levých hran.
- Uživatel má tendenci číst rozhraní zleva doprava.



Obr. 3.8: Prototyp GUI pro případ užití AktualizovatData o osvětlení pro aktéra technik.

Po vyhledání prvku lze podle potřeby upravit data o osvětlení (tabulka VO_VEROSV_GS) a svítidlech (VO_SVIT), nebo lokalizovat prvek v programu GS Web. Vazba mezi tabulkou osvětlení VO_VEROSV_GS a tabulkou VO_SVIT je 1:N. Tudíž jedno osvětlení může mít několik svítidel.

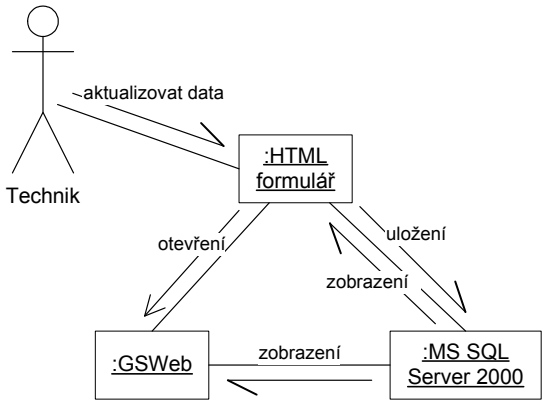


Obr. 3.9: Prototyp GUI pro aktualizaci osvětlení a svítidlech pro aktéra technik.

3.1.9 Diagram spolupráce

Diagram spolupráce je zjednodušený oproti sekvenčnímu o časové informace, díky čemuž dokážeme lépe pochopit předávání zpráv mezi objekty. Diagram sekvencí a diagram spolupráce jsou sémanticky ekvivalentní. Zatímco diagram sekvencí popisuje to, co se děje v čase, tak diagram spolupráce popisuje co se děje v prostoru, proto při vývoji systému bude použit diagram spolupráce.

Pro nového aktéra *technik* byl taktéž vytvořen nový diagram spolupráce, podobný diagramu spolupráce, který popisuje práci aktéra *pracovník EM*. Rozdíl je pouze v GUI, pomocí něhož aktér komunikuje s databází a programem GS Web. *Technik* bude pracovat s poměrně jednoduchým GUI ve formě *HTML formuláře*, ve kterém bude aktualizovat data.



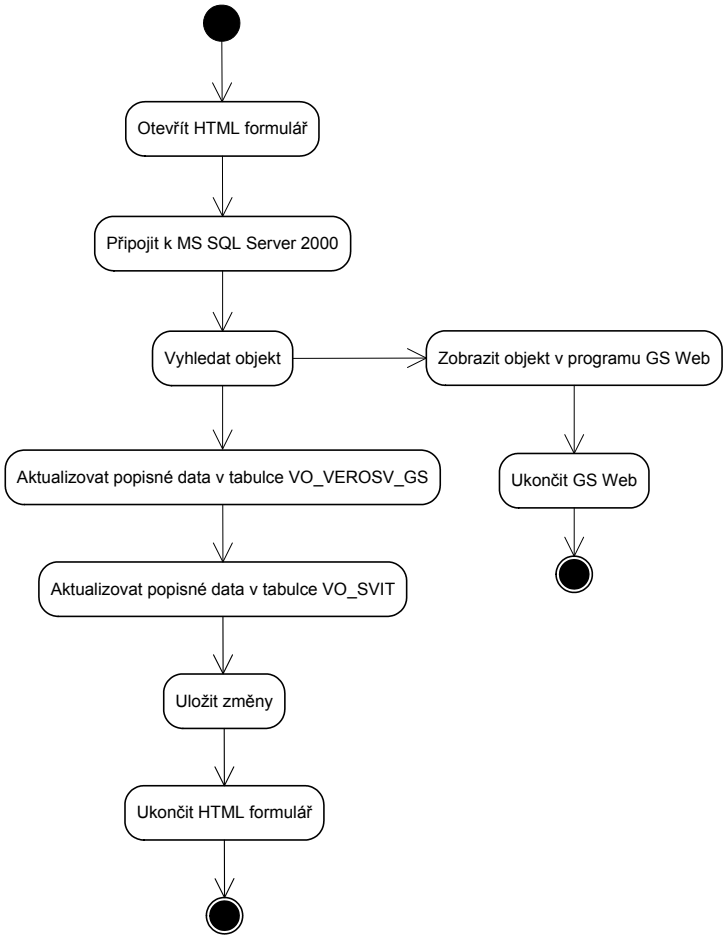
Obr. 3.11: Diagram spolupráce pro aktéra technik.

Popis sekvencí:

- *Technik* aktualizuje data v HTML formuláři.
- HTML formulář ukládá data do databáze, která je řízená databázovou platformou MS SQL Server 2000.
- Pro zobrazení prostorových dat otevře HTML formulář prohlížeč GS Web.

3.1.10 Diagram činnosti

Diagram činností patří mezi základní vývojové diagramy, které jsou jedny z prvních vizuálních modelů, které se v programování objevily. Díky tomu, že zobrazují pořadí kroků, jsou pro laika lehce pochopitelné. Navíc diagram zobrazuje i body v nichž dochází k rozhodování a program se v nich větví. Diagram vzniká rozšířením stavového diagramu. Na obr.3.8 je znázorněn stav *práce* pro aktéra *technik*.



Obr. 3.12: Diagram činností popisující práci technika

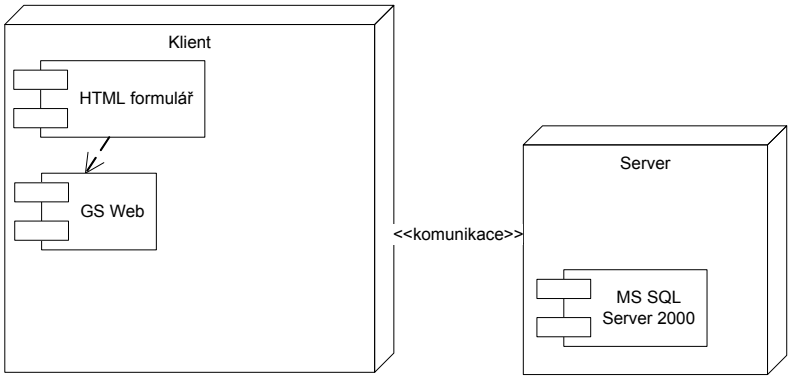
Modelujeme proces, k níž dochází během případu užití *AktualizovatData* pro nového aktéra *technik* a které je potřeba vykonat.

Popis činnosti *technika*:

- Otevře HTML formulář.
- Připojí se k databázi.
- Vyhledání objektu veřejného osvětlení pomocí jednoznačného identifikátoru *id_misto*.
- Jestliže chce objekt zobrazit potom otevře GS Web.
- Jinak aktualizuje atributová data v nadřízené tabulce VO_VEROSV_GS.
- Aktualizuje atributová data v podřízené tabulce VO_SVIT.
- Uloží změny.
- Ukončí HTML formulář, popř. GS Web.

3.1.12 Diagram komponent

Softwarová komponenta *HTML formulář* bude fyzická součást systému, který bude používat t*echnik*. *Technik* bude zároveň používat *GS We*b, který se bude spouštět automaticky z *HTML formuláře* při dotazu na lokalizaci objektu. Jejich vztah je takový, že komponenta *HTML formulář* volá komponentu *GS Web*. Celý tento funkční systém klienta bude komunikovat se serverem, kde je uložená komponenta *MS SQL Server 2000*, která tvoří databázovou platformu.



Obr. 3.13: Diagram komponent a jejich nasazení v optimalizovaném systému.

3.2 Diskuze

Diskuse splnění cílů:

- *Analýza GIS města Plzně* byla splněna v 1. kapitole. Analýza posloužila zejména pro SWOT analýzu současného stavu klientů GIS, především pro určení příležitostí a hrozeb.
- *Analýza systému užití klientů GIS na SVSmP* splněna v 2. kapitole. Analýza posloužila zejména pro SWOT analýzu současného stavu klientů GIS, především pro určení silných a slabých stránek.
- *Popis metodiky vývoje systému* pomocí modelovacího jazyka UML byl splněn v 2. kapitole. Ve 3. kapitole byla pak úspěšně vyzkoušena na vývoji klientů GIS.
- *Zkvalitnění využívání geografických informací* bylo splněno v 3. kapitole. Cíl byl splněn vhodným přiřazením *technika* k aktualizaci dat, tvorbou datového modelu podle přání uživatelů a zároveň databázových pravidel a v neposlední řadě tvorbou jednoduchého klienta GIS pro práci *technika*.

3.2.1 Tvorba a testování GUI

Pro účely testování byl zvolen software:

- Operační systém Windows XP,
- databázová platforma MySQL 5.0.18,
- skriptovací jazyk PHP 5.2.1,
- univerzální rozhraní ODBC,
- ovladač MyODBC 3.51.12,
- HTTP server Apache 2.0.59.

MySQL je jeden z nejrozšířenějších Open Source systémů pro řízení dat. V této databázové platformě byla vytvořena kopie databáze VO podle datového modelu z obr. 3.5. a nahrána testovací data. SQL příkaz pro vytvoření databáze a nahrání dat viz. Příloha B. Dále byl

vytvořen jednoduchý HTML formulář (viz. Příloha C), kde se o dynamické generování starají příkazy skriptu napsaném v jazyce PHP. Vyhodnocení a zpracování příkazů provádí HTTP server Apache. HTML formulář byl propojen s databází pomocí ODBC. Velkou výhodou ODBC je, že stejným způsobem můžeme přistupovat k libovolné databázi. Pokud se tedy změní SQL-server, na kterém běží naše aplikace, nemusíme měnit v PHP-skriptech žádný kód. Stačí upravit mapování datových zdrojů na databáze v konfiguračním programu ODBC. Databázová platforma MySQL podporuje ODBC, díky řadiči MyODBC. Testování proběhlo v pořádku a klient GIS bude moci být nasazen do ostrého provozu.

4. Závěr

Diplomová práce analyzovala GIS města Plzně z pohledu jejího provozovatele (SITmP) a zároveň z pohledu organizace, která je jedním z hlavních zdrojů geodat pro tento GIS (SVSmP). Byly popsány hlavní technologie, které používá SVSmP pro práci s geodaty a uživatelé technologií. Pro objektivnost byla vytvořena SWOT analýza poukazující na silné a slabé stránky se zaměřením na interní prostředí SVSmP a příležitosti a hrozby se zaměřením na externí prostředí této organizace. Dále byla vytvořena metodika vývoje systému. Teoretická část rozebírala náročnou problematiku pomocí diagramů UML, čímž se čtenáři dostala příležitost seznámit se s tímto modelovacím jazykem. Pro názorný příklad byla vybrána problematika evidence VO města Plzně.

Z analytických kroků vyplynula možnost zefektivnění práce, díky přiřazení jednoduchého klienta GIS *technikům* VO, kteří se doposud zadáváním dat do databáze nezabývali. Možnost byla zrealizována v praktické části, kde navíc byl vytvořen nový datový model, a to podle standardů UML již popsané v předchozí části. Diplomovou prací aktivity autora nekončí. Datový model byl již do geodatabáze města Plzně implementován, nicméně klient GIS na příležitost ostrého provozu jistě čeká. Celý proces se v duchu tohoto projektu může opakovat i u ostatního evidovaného majetku.

5. Slovník použitých termínů a zkratek

Databáze – prostor, který obsahuje data nebo metadata.

Databáze, relační – databázový systém, který je založen na relačním modelu dat a relační algebře. Data jsou uspořádána do tabulek (relací), nad kterými jsou definovány přípustné operace.

Databázová platforma – konkrétní databázový programový produkt, např. MS SQL Server 2000, MS SQL Server 2005, Oracle Database 10g, Sybase ASE 15, IBM DB2.

CAD – Computer Aided Design, počítačem podporované projektování nebo počítačem podporovaný návrh je používání pokročilých počítačových programů při projektování místo rýsovacího prkna.

CGI – Common Gateway Interface, protokol pro propojení externí aplikace s webovým serverem.

EM – Evidence majetku, oddělení Správy veřejného statku města Plzně.

Geodata – Geografická data, propojená atributová a prostorová data.

GIS – Geografický informační systém, slouží k analýze a modelování již existujícího světa, včetně historie.

GPS – Global Positioning System je vojenský navigační družicový systém provozovaný Ministerstvem obrany Spojených států amerických, který dokáže s několikametrovou přesností určit pozici kdekoliv na Zemi.

GUI – Graphical User Interface nebo-li grafické uživatelské rozhraní, druh komunikace s počítačem mající podobu interaktivních prvků.

HTML – Hyper Text Markup Language. Je jedním z jazyků pro vytváření stránek v systému World Wide Web, který umožňuje publikaci stránek na Internetu.

HTTP – Hyper Text Transfer Protocol. Je internetový protokol určený původně pro přenos hypertextových dokumentů ve formátu HTML.

Java – objektově orientovaný programovací jazyk, který vyvinula a roku 1995 představila firma Sun Microsystem. Je jedním z nejpoužívanějším programovacích jazyků na světě.

Klient/server architektura – síťová architektura, která zvyšuje výkonnost tím, že odděluje klienta a server.

Komponenta – spustitelná část kódu, která je schopná spojení s jinými komponentami.

Metadata – strukturovaná data o datech. Metadata mohou sloužit např. k snadnému vyhledávání.

MISNet – Optická síť zajišťující datové propojení mezi objekty.

ODBC – Open DataBase Connectivity, mezivrstva pro obsluhu databáze.

OGC – mezinárodní neziskové konsorcium Open Geospatial Consorcium, Inc.

OMG – Object Management Group, standardizační komise.

Optimalizace – proces výběru nejlepší varianty z množství možných jevů.

PDA – Personal Digital Assistant, nebo-li osobní digitální pomocník je malý kapesní počítač, který je ovládaný obvykle dotykovou obrazovkou a perem (tzv. stylem).

PHP – skriptovací jazyk.

RDBMS – Relational Database Management System. Jde o software pro řízení relační databáze. Někdy se také používá český překlad tohoto označení – Systémy řízení bází dat neboli SŘBD.

SITmP – Správa informačních technologií města Plzně.

Server – je v informatice obecné označení pro proces nebo systém, který poskytuje nějakou službu. Jako server se také označuje samotný počítač, který tyto činnosti vykonává.

Server, aplikační – počítač specializovaný na provoz nějaké aplikace.

Server, databázový – slouží jako úložiště strukturovaných dat (databází) a umožňuje provádět jejich efektivní vyhledávání.

Server, webový – počítač připojený k síti Internet, který poskytuje www stránky.

SVSmP – Správa veřejného statku města Plzně.

SWOT – Strengths Weaknesses Opportunities Threats, komplexní metoda kvalitativního vyhodnocení.

UML – Unified Modeling Language, univerzální jazyk pro vizuální modelování systému.

Benchmarking – měření a analýza procesů a výkonů organizace a hledání nejlepších řešení prostřednictvím systematického porovnávání s výkonem ostatních.

WMS – Web Map Service, protokol umožňující přístup klientů k mapovému serveru, který je na Internetu.

.NET – zkráceně Network, nebo-li síť. Zastřešující název pro soubor technologií v softwarových produktech, které tvoří celou platformu dostupnou pro Web, Windows a PocketPC.

.NET Compact Framework – verze .NET platformy pro Pocket PC s operačním systémem Windows Mobile, která je s klasickou verzí kompatibilní a tak není nutné kompilovat různé aplikace pro PC a PDA - na obou systémech bude aplikace fungovat stejně.

6. Zdroj informací

[1] ARLOW, J.- NEUSTADT, I. *UML a unifikovaný proces vývoje aplikací.* Přeložil Bohdan Kostka. 1.vyd. Brno: Computer Press, 2003. ISBN 80-7226-947-X.

[2] FRIEDRICH, V. - LUKÁŠ, M. *Informační systémy veřejné správy.* Plzeň: Západočeská univerzita, 1999. ISBN 80-7082-555-3.

[3] GeoBusiness. *Č.1/2007*. Praha: Springwinter, 2007. Vychází čtvrtletně. ISSN 1214-220.

[4] GEOVAP spol. s r.o. GS technologie [on-line]. c2001-2006, [cit. 2006-09-29]. URL: http://www.geostore.cz/index.asp

[5] Help Servise-remote sensing spol. s r.o. Webové služby [on-line]. [cit. 2007-04-04]. URL: http://www.bnhelp.cz/bnhelp/ows.htm

[6] HNOJIL, J. Geoinformatika a veřejná správa v 21. století. *GEOinformace*, zima-jaro 2006, s. 28-29. ISSN 1214-220.

[7] JEDLIČKA, K. *Podkladová práce k diskusi o pojmech modelování a simulace* [on-line]. Plzeň: ZČU, 2006. [cit. 2007-02-20].

[8] KANISOVÁ, H.- MŰLLER, M., *UML srozumitelně.* 1.vyd. Brno: Computer Press, 2004. ISBN 80-251-0231-9.

[9] KEOGH, J.- GIANNINI, M.*OOP bez předchozích znalostí.* Přeložila Matějů Veronika. 1.vyd. Brno: Computer Press, 2006. ISBN 80-251-0973-9.

[10] KUTÁČ, D. Databázový svět. Jak objektové databáze podporují škálovatelnost poprvé[on-line]. [cit. 2006-11-06]. URL: http://www.dbsvet.cz/view.php?cisloclanku=2006051501

[11] MAREŠ, S. - MOHELSKÁ, H. – ŠABATOVÁ, M. *Manažerské metody.* Hradec Králové: Gaudeamus, 2004. ISBN 80-7041-534-7.

[12] NOVOTNÝ, J. *Informační systém malé obce.* (Bakalářská práce) Plzeň: ZČU, 2005. Vedoucí bakalářské práce Ing. Mgr. Otakar Čerba.

[13] PAGE-JONES, M. *Základy objektově orientovaného návrhu v UML.* Přeložil Voráček Karel. 1.vyd. Praha: Grada, 2002. ISBN 80-247-0210-X.

[14] POKORNÝ, M. Databázový svět. Vyvíjíme databázový a informační systém XXIII. [on-line]. [cit. 2007-01-19]. URL: http://www.dbsvet.cz/rservice.php?akce=tisk&cisloclanku=2004121602&button=no

[15] SCHMULLER, J. *Myslíme v jazyku UML.* 1.vyd. Praha: Grada, 2001. ISBN 80-247-0029-8.

[16] Správa informačních technologií města Plzně. Geografický informační systém města Plzně[on-line].c2002, [cit. 2006-09-29]. URL: http://gis.plzen-city.cz/ogis

[17] Stavební technologie. SWOT analýza [on-line]. c2001-2004, [cit. 2006-10-12]. URL: http://www.stavebnitechnologie.cz/view.php?cisloclanku=2002041701

[18] ŠÍMA, J. *Geoinformační terminologie pro geodety a kartografy*. Zdiby: VÚGTK, 2003. 37 s. a 41 s. ISBN 80-85881-20-9.

[19] TALICH, M. *Webové služby a aplikace XML* [on-line]. c2004, [cit. 2007-04-04]. URL: http://www.inforum.cz/inforum2004/pdf/Talich_Milan.pdf

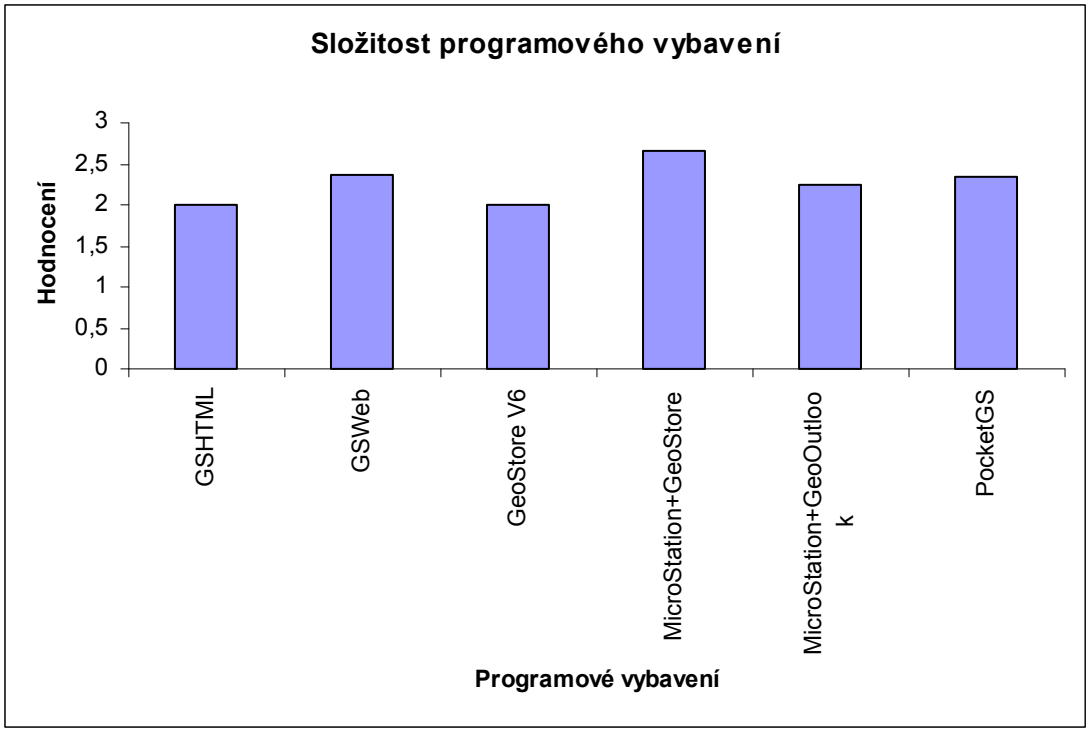
[20] TopoL Software, spol. s r.o. Co je WMS [on-line].c1999-2006,[cit.2007-01-15]. URL: http://www.topol.cz/?doc=3040

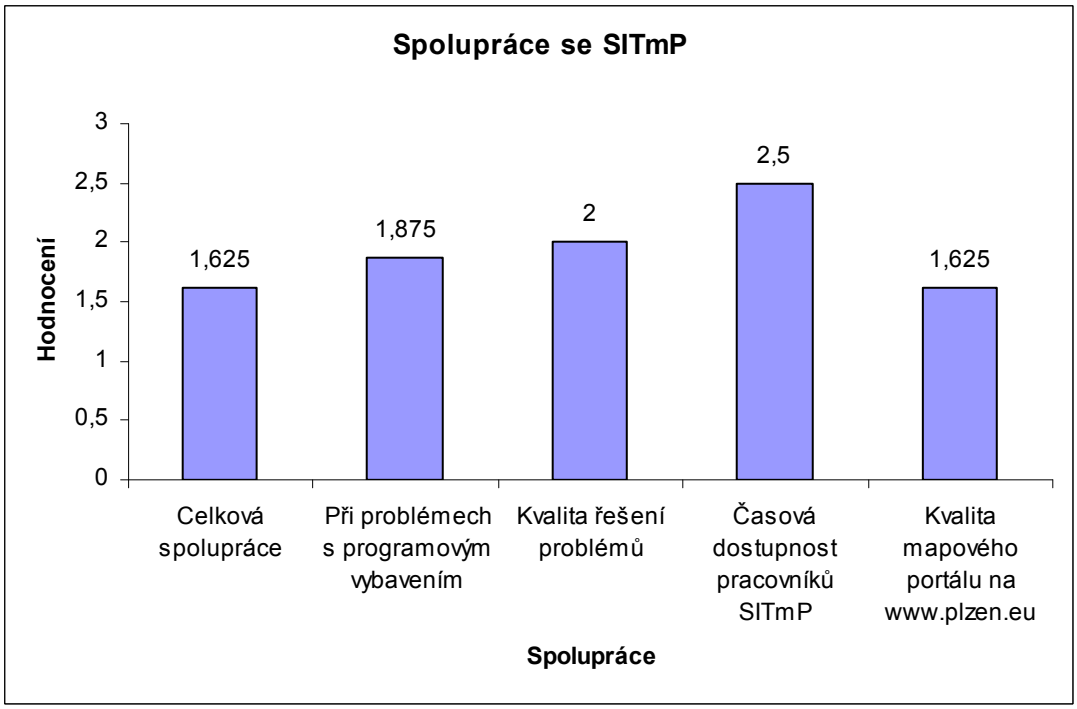
[21] TUČEK, J. *Geografické informační systémy- principy a praxe*. Praha: Computer Press, 1998. ISBN 80-7226-091-X.

[22] VOKOUNOVÁ, L. Návrh struktury datového modelu pro správu elektrických distribučních sítí ZČE v GIS analýzou mezinárodního datového modelu ArcFM. (Diplomová práce) Plzeň: ZČU, 2003. Vedoucí diplomové práce Ing. Karel Jedlička.

[23] ŽAMPA, P. Základy teorie systémů [on-line].Plzeň: ZČU. [cit. 2007-02-20]. URL: http://control.zcu.cz/predmety/ts/tsprac.pdf

7. Příloha A: Anketa a zpracování





Pozn. Interval 1 až 5, 1 znamená intuitivní program, 5 znamená složitý program.

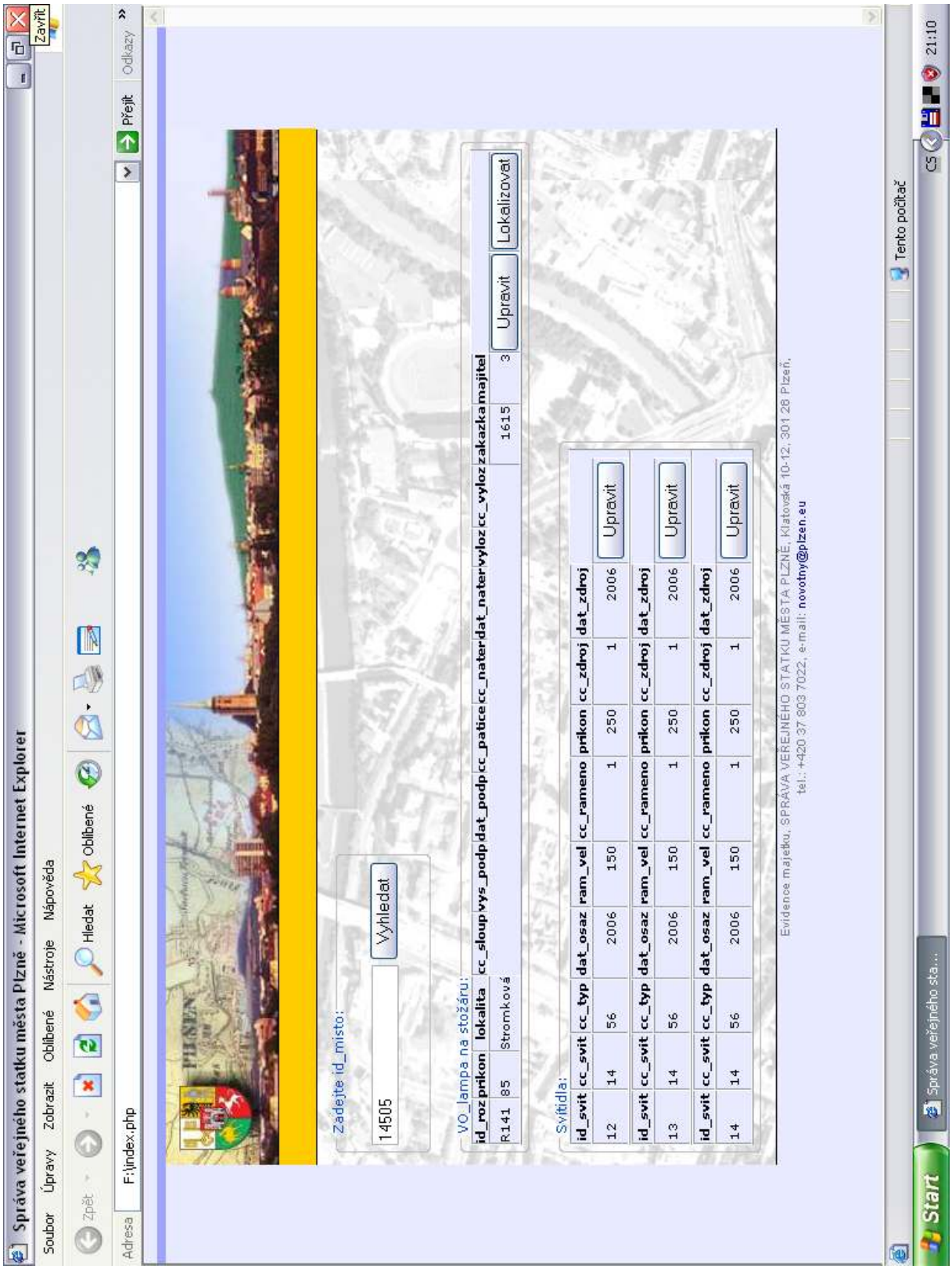
8. Příloha B: Skript SQL pro vytvoření a naplnění databáze

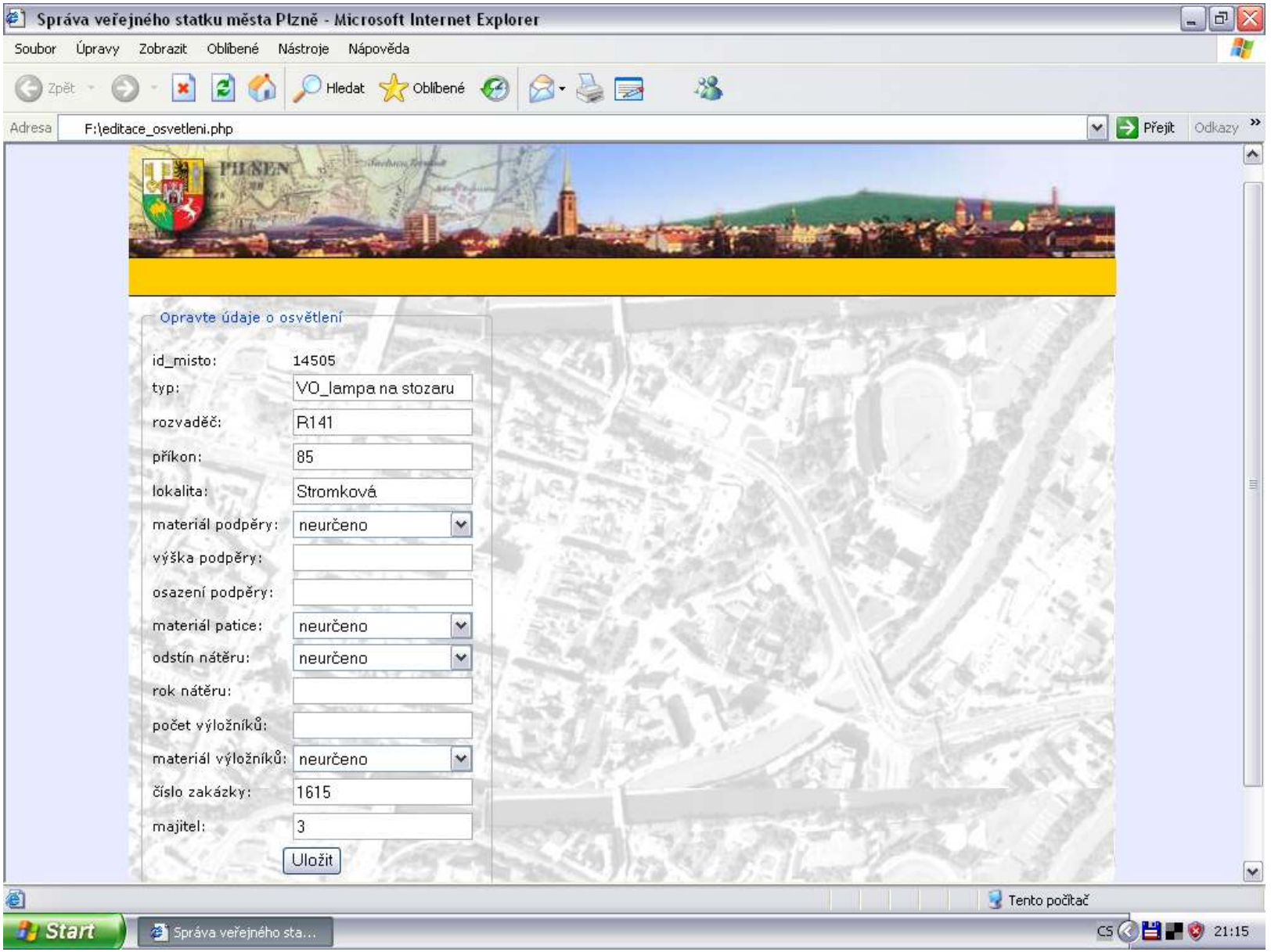
```
#
# Spuštěním tohoto skriptu na SQL serveru se vygenerují dvě #
#tabulky, které využívají připojené skripty v jazyce PHP
#
# MySQL dump 5.0
#
# Host: localhost    Database: VerejneOsvetleni
#--------------------------------------------------------

#
# Table structure for table 'vo_verosv_gs'
#
CREATE DATABASE VerejneOsvetleni;
CREATE TABLE vo_verosv_gs (
     id_misto int NOT NULL,
     rc varchar(20),
     id_roz varchar(5),
     prikon int,
     lokalita varchar(30),
     cc_sloup int,
     vys_podp int,
     dat_podp int,
     cc_patice int,
     cc_nater int,
     dat_nater int,
     vyloz int,
     cc_vyloz int,
     zakazka int,
     majitel int,
     CONSTRAINT pk_verosv PRIMARY KEY (id_misto)
               ON DELETE CASCADE
);
```

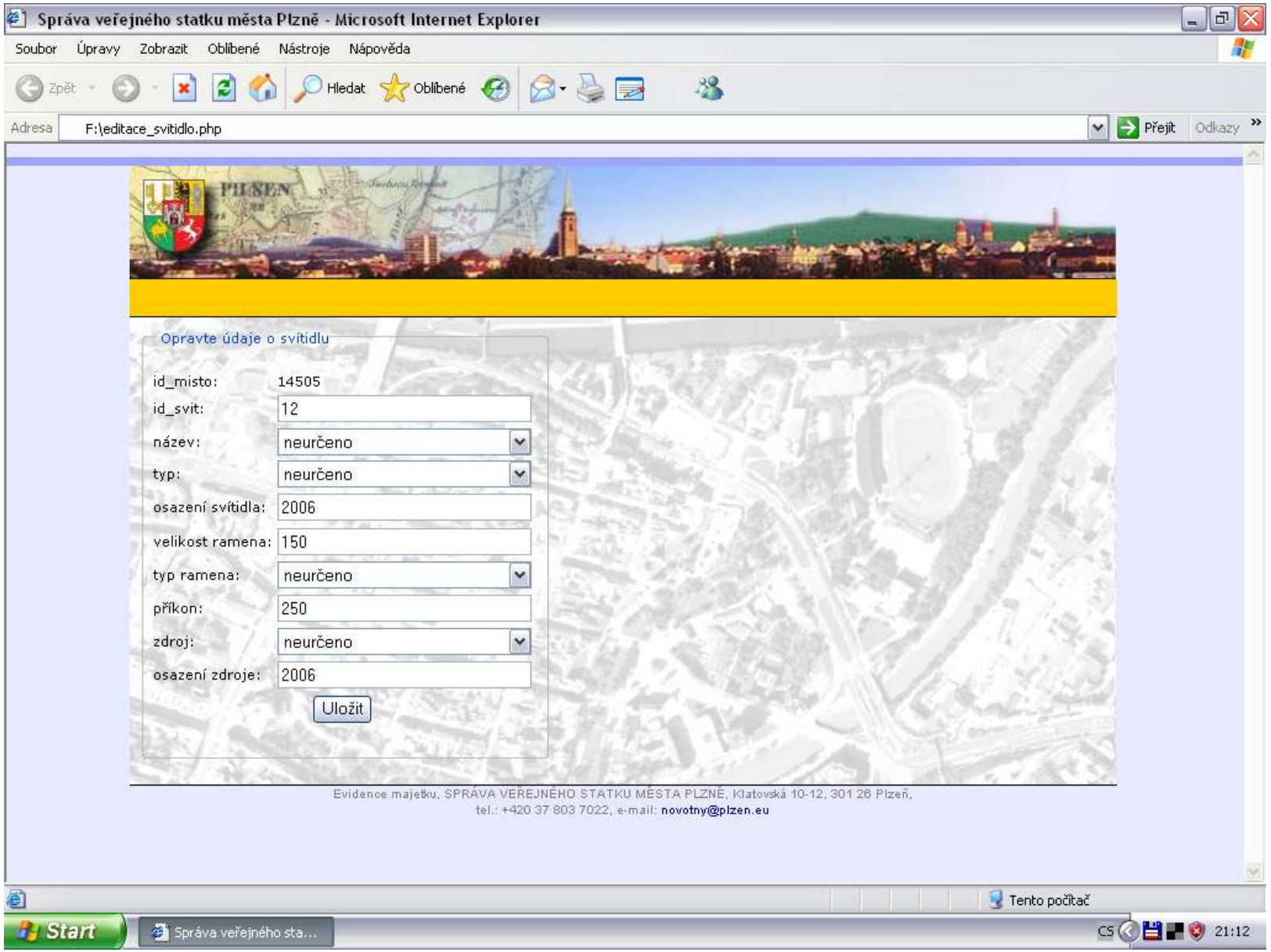
```
#
# Dumping data for table 'vo_verosv_gs'
#

INSERT INTO vo_verosv_gs (id_misto, rc, id_roz, prikon,
lokalita, zakazka, majitel) VALUES (14505,'VO_lampa na
stožáru','R141',85,'Stromková',1615,3);
INSERT INTO vo_verosv_gs (id_misto, rc, id_roz, prikon,
lokalita, zakazka, majitel) VALUES (14517,'VO_lampa na
stožáru','R141',85,'Radčice',1615,3);
INSERT INTO vo_verosv_gs (id_misto, rc, id_roz, prikon,
lokalita, zakazka, majitel) VALUES (13308,'VO_lampa na
stožáru','R106',170,'Manětínská',1,3);
INSERT INTO vo_verosv_gs (id_misto, rc, id_roz, prikon,
lokalita, zakazka, majitel) VALUES (12797,'VO_lampa na
stožáru','R111',170,'Tachovská',1615,3);
INSERT INTO vo_verosv_gs (id_misto, rc, id_roz, prikon,
lokalita, zakazka, majitel) VALUES (12320,'VO_lampa na
stožáru','R111',85,'Jesenická',1615,3);

#
# Table structure for table 'vo_verosv_gs'
#

CREATE TABLE vo_svit (
     id_svit int NOT NULL,
     cc_svit int,
     cc_typ int,
     dat_osaz int,
     ram_vel int,
     cc_rameno int,
     prikon int,
     cc_zdroj int,
```

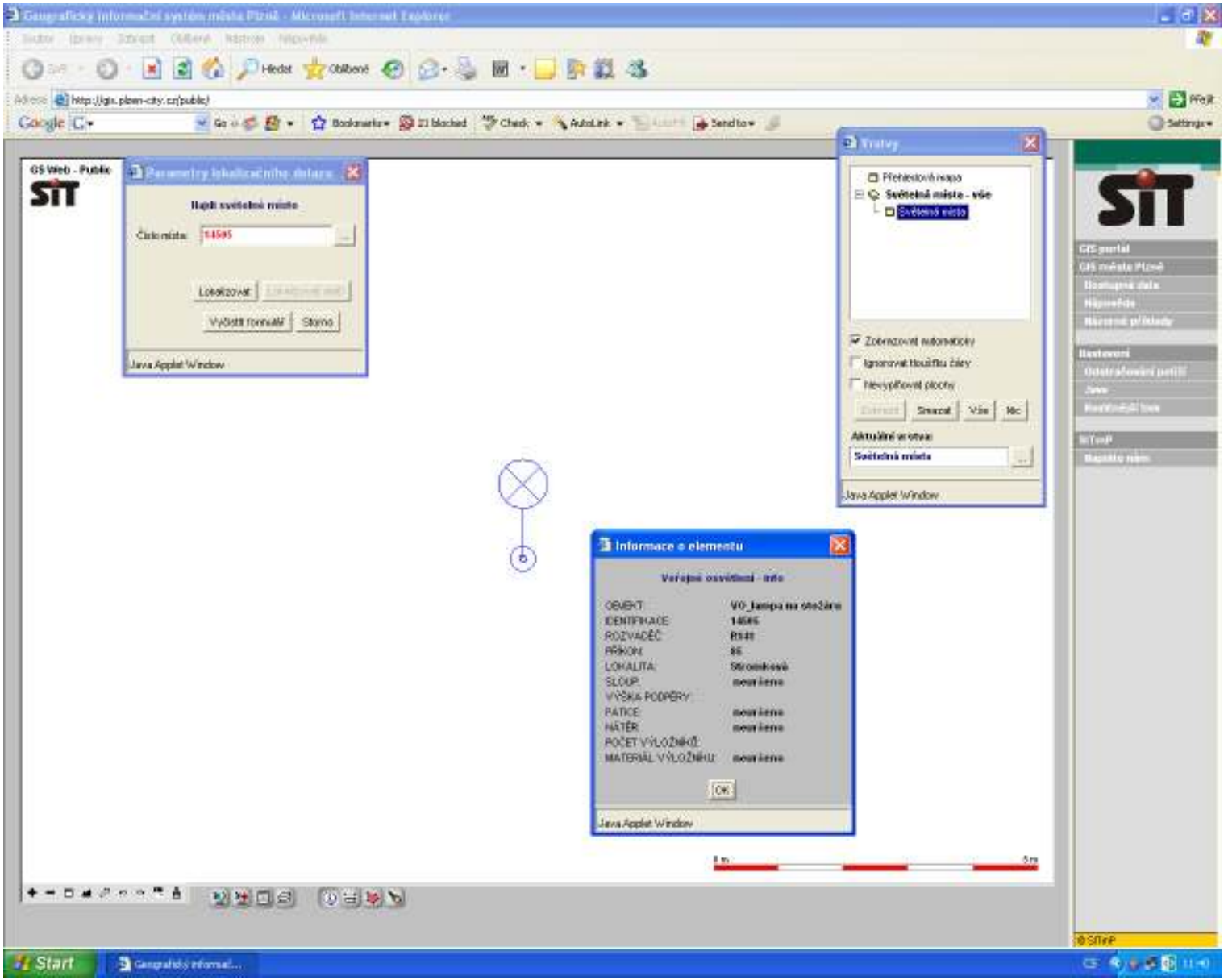
```
     dat_zdroj int,
     id_misto int,
     CONSTRAINT pk_svit PRIMARY KEY (id_svit),
     CONSTRAINT fk_svit FOREIGN KEY (id_misto)
                         REFERENCES vo_verosv_gs (id_misto)
);

#
# Dumping data for table 'vo_verosv_gs'
#

INSERT INTO vo_svit (id_misto, id_svit, cc_svit, cc_typ,
dat_osaz, ram_vel, cc_rameno, prikon, cc_zdroj, dat_zdroj)
VALUES (14505,12,14,56,2006,150,1,250,1,2006);
INSERT INTO vo_svit (id_misto, id_svit, cc_svit, cc_typ,
dat_osaz, ram_vel, cc_rameno, prikon, cc_zdroj, dat_zdroj)
VALUES (14505,13,14,56,2006,150,1,250,1,2006);
INSERT INTO vo_svit (id_misto, id_svit, cc_svit, cc_typ,
dat_osaz, ram_vel, cc_rameno, prikon, cc_zdroj, dat_zdroj)
VALUES (14505,14,14,56,2006,150,1,250,1,2006);
INSERT INTO vo_svit (id_misto, id_svit, cc_svit, cc_typ,
dat_osaz, ram_vel, cc_rameno, prikon, cc_zdroj, dat_zdroj)
VALUES (14517,15,14,56,2006,150,1,250,1,2006);
INSERT INTO vo_svit (id_misto, id_svit, cc_svit, cc_typ,
dat_osaz, ram_vel, cc_rameno, prikon, cc_zdroj, dat_zdroj)
VALUES (13308,16,20,56,2006,150,0,250,1,2006);
INSERT INTO vo_svit (id_misto, id_svit, cc_svit, cc_typ,
dat_osaz, ram_vel, cc_rameno, prikon, cc_zdroj, dat_zdroj)
VALUES (12797,17,14,25,2007,150,1,100,1,2007);
INSERT INTO vo_svit (id_misto, id_svit, cc_svit, cc_typ,
dat_osaz, ram_vel, cc_rameno, prikon, cc_zdroj, dat_zdroj)
VALUES (12320,18,2,0,2007,0,1,250,1,2006);
```

9. Příloha C: Ukázka návrhu jednoduchého klienta GIS




Správa veřejného statku města Plzně - Microsoft Internet Explorer
Soubor Úpravy Zobrazit Oblíbené Nástroje Nápověda
Zpět Hledat Oblíbené
Adresa F:\editace_osvetleni.php
Přejít Odkazy
PLZEŇ
Opravte údaje o osvětlení
id_misto: 14505
typ: VO_lampa na stozaru
rozvaděč: R141
příkon: 85
lokalita: Stromková
materiál podpěry: neurčeno
výška podpěry:
osazení podpěry:
materiál patice: neurčeno
odstín nátěru: neurčeno
rok nátěru:
počet výložníků:
materiál výložníků: neurčeno
číslo zakázky: 1615
majitel: 3
Uložit
Tento počítač
Start
Správa veřejného sta...
CS
21:15


Správa veřejného statku města Plzně - Microsoft Internet Explorer
Soubor Úpravy Zobrazit Oblíbené Nástroje Nápověda
Zpět Hledat Oblíbené
Adresa F:\editace_svitidlo.php
Přejít Odkazy
PLZEŇ
Opravte údaje o svítidlu
id_misto: 14505
id_svit: 12
název: neurčeno
typ: neurčeno
osazení svítidla: 2006
velikost ramena: 150
typ ramena: neurčeno
příkon: 250
zdroj: neurčeno
osazení zdroje: 2006
Uložit
Evidence majetku, SPRÁVA VEŘEJNÉHO STATKU MĚSTA PLZNĚ, Klatovská 10-12, 301 26 Plzeň,
tel.: +420 37 803 7022, e-mail: novotny@plzen.eu
Tento počítač
Start
Správa veřejného sta...
CS
21:12


Geografický informační systém města Plzně - Microsoft Internet Explorer
GIS Web - Public
SIT
Parametry lokalizačního dotazu
Najdi světelné místo
Číslo místa:
Lokalizovat
Vyčistit formulář
Storno
Java Applet Window
Trasy
Přehledová mapa
Světelná místa - vše
Světelná místa
Zobrazovat automaticky
Ignorovat tloušťku čáry
Nevyplňovat plochy
Smazat
Vše
Nic
Aktuální vrstva:
Světelná místa
Java Applet Window
Informace o elementu
Veřejné osvětlení - info
OBJEKT
IDENTIFIKACE
ROZVADĚČ
PŘÍKON
LOKALITA
SLOUP
VÝŠKA PODPĚRY
PATICE
NÁTĚR
POČET VÝLOŽNÍKŮ
MATERIÁL VÝLOŽNÍKU
OK
Java Applet Window
GIS portál
GIS města Plzně
Nastavení
SITinP
Start