

Západočeská univerzita v Plzni

Fakulta aplikovaných věd

Katedra matematiky



## **Diplomová práce**

# **Open source mapový server pro data katastru nemovitostí**

Plzeň 2007

Jiří Petrák

Předkládám k posouzení a obhajobě diplomovou práci, zpracovanou na závěr studia na Fakultě aplikovaných věd Západočeské univerzity v Plzni.

Prohlašuji, že jsem diplomovou práci vypracoval samostatně a výhradně s použitím citovaných pramenů.

V Plzni dne 24. května 2007

Jiří Petrák

Na tomto místě bych rád poděkoval všem, kteří přispěli k realizaci této práce, zejména Ing. Karlu Jedličkovi za ochotu, cenné nápady, rady a připomínky. Poděkování patří i Ing. Karlu Vondráčkovi ml. a firmě Georeal spol. s r.o. za prvotní podnět k realizaci této práce. Děkuji rovněž své rodině za trpělivost a podporu ve studiu.

# **Abstrakt**

Diplomová práce se zabývá publikací dat informačního systému katastru nemovitostí České republiky v prostředí webu. Jako teoretický základ je obecně nastíněna architektura řešení mapového serveru a popsány jednotlivé komponenty. Druhá část se věnuje výměnnému formátu informačního systému katastru nemovitostí České republiky a zaměřuje se především na způsob získání atributových informací o nemovitostech, nutných k sestavení listů vlastnictví. Tyto poznatky byly prakticky využity k vytvoření aplikace, postavené výhradně na nekomerčním software. Způsobem implementace této aplikace se zabývá třetí část práce.

# **Abstract**

The thesis occupies with publishing the data of the Czech Republic Land Registry Information System on the Web. As a theoretic basics, a general architecture of a map server solution is outlined and the components described individually. The second section engages in the Land Registry Information System exchange format and first of all concentrates on the way of real estate attribute information acquisition, that is necessary for the Ownership Certificate construction. This knowledge was utilized to build a map application, entirely based on non-commercial software. The third section of the thesis deals with this application implementation.

# Obsah

1. Úvod .....	7
2. Architektura řešení mapového serveru .....	8
2.1. Rozdělení GIS .....	8
2.2. Součásti internetového GIS .....	9
2.2.1. Klient .....	9
2.2.2. Webový a aplikační server .....	14
2.2.3. Mapový server .....	15
2.2.4. Datový server .....	15
2.3. Interoperabilita v GIS podle OGC .....	15
2.3.1. Open Geospatial Consortium .....	16
2.3.2. Webové služby specifikované OGC .....	16
3. Popis struktury výměnného formátu informačního systému katastru nemovitostí České republiky .....	18
3.1. Části VFK .....	18
3.1.1. Hlavička .....	18
3.1.2. Datové bloky .....	19
3.2. Popis vybraných skupin datových bloků .....	21
3.2.1. Nemovitosti (NEMO) .....	21
3.2.2. Vlastnictví (VLST) .....	23
3.2.3. Jiné právní vztahy (JPVZ) .....	24
3.2.4. Řízení (RIZE) .....	25
3.2.5. Bonitní díly parcel (BDPA) .....	26
3.3. Parcelní a domovní čísla ve VF ISKN .....	27
4. Implementace zvoleného mapového serveru pro katastrální data .....	28
4.1. Konkrétní řešení mapového serveru .....	28
4.2. Instalace PostGIS a import dat VF ISKN .....	30
4.3. Instalace mapového a webového serveru .....	31
4.3.1. MapScript vs. CGI .....	32
4.4. Vizualizace prostorových dat .....	32
4.4.1. Mapsoubor .....	33
4.4.2. Stránka s mapou .....	35
4.5. Základní atributové informace nemovitostí .....	37
4.6. Sestavení listu vlastnictví .....	39
4.6.1. Záhloví .....	39
4.6.2. Část A .....	40

---

4.6.3. Část B .....	42
4.6.4. Části B1, C, D .....	43
4.6.5. Část E .....	47
4.6.6. Část F .....	48
5. Závěr .....	50
Literatura .....	51
1. Grafické ukázky z aplikace .....	53
2. Obsah přiloženého CD .....	56

# Kapitola 1

## Úvod

S rozvojem Internetu a počítačových sítí obecně dochází ke stále většímu rozšíření klient-server aplikací. Tento trend se nevyhnul ani oblasti GIS. Klient-server aplikace mají oproti izolovaným desktopovým řešením řadu výhod: představují snadný a rychlý přístup k informačním technologiím bez nutnosti instalace specializovaného softwaru, odpadá nutnost pořizování dat, umožňují lepší organizaci práce, snadnější správu, aktualizaci apod.

Data katastru nemovitostí patří k často využívaným prostorovým datům. Například pro správní orgány, jako jsou městské a obecní úřady, je přístup k datům katastru nemovitostí nezbytnou záležitostí - potřebují je pro rozhodování, ale i běžnou administrativu a usnadnění běžného života obyvatel.

V diplomové práci J. Orálka [Ora2006] byla vyřešena záležitost importu dat z výměnného formátu informačního systému katastru nemovitostí do prostorové databáze PostGIS. V této fázi je tedy možné tato data vizualizovat s použitím některého silného klienta. Moje práce si klade za cíl navrhnout a implementovat klient-server řešení, které by umožnilo prohlížet tato data pomocí tenkého klienta (webového prohlížeče). Mapový server by měl umožňovat základní práci s mapou (posun, přiblížení, oddálení, vypínání vrstev) a především získávání atributových informací o nemovitostech v rozsahu, který by umožňoval sestavit Výpis z katastru nemovitostí („List vlastnictví“).

# Kapitola 2

## Architektura řešení mapového serveru

Aplikace běžně nazývané mapovými servery jsou vlastně distribuovanými *geografickými informačními systémy (GIS)* s poněkud omezenou funkcí - jsou zaměřené hlavně na zobrazování geodat, případně jednoduché dotazy. Mapový server je pak jejich nejdůležitější součástí, běžící na straně serveru, zodpovědná za všechnu GIS-funkcionalitu.

*Geografický informační systém (GIS)* je organizovaný souhrn počítačové techniky, programového vybavení, geografických dat a zaměstnanců navržený tak, aby mohl efektivně získávat, ukládat, aktualizovat, analyzovat, přenášet a zobrazovat všechny druhy geograficky vztahovaných informací [i-ARC].

*Geografické informace* jsou informace o jevech, jejichž poloha je přímo nebo nepřímo vztahována k povrchu Země [TSG]. *Geodata* jsou potom formální popisy (přepis) geoinformace ve formě čísel a znaků vhodné pro počítačové zpracování [Tuč1998].

### 2.1. Rozdělení GIS

Vývoj GIS je závislý na vývoji informačních technologií a v podstatě ho kopíruje. Rozvoj internetu tedy významně ovlivnil i oblast GIS. Vývoj GIS se z hlediska využití sítí dá rozdělit na 3 základní fáze (dle [Pen2003]):

- *Mainframe GIS* - programy na střediskových a sálových počítačích s terminálovým přístupem (záležitost minulosti, nyní po spojení s *desktop GIS* přecházejí v *distribuované GIS*).
- *Desktop GIS* - programy na stolních počítačích, buď samostatné, kde nedochází k žádné komunikaci, nebo spojené v rámci Local Area Network (LAN) (minulost a současnost).
- *Distribuované GIS* (distributed GIS) - uživatel zpravidla nepotřebuje speciální program na svém koncovém zařízení, přístup k datům i funkcím má prostřednictvím sítě (současnost a budoucnost). Distribuované GIS se dále dělí na dvě podskupiny:



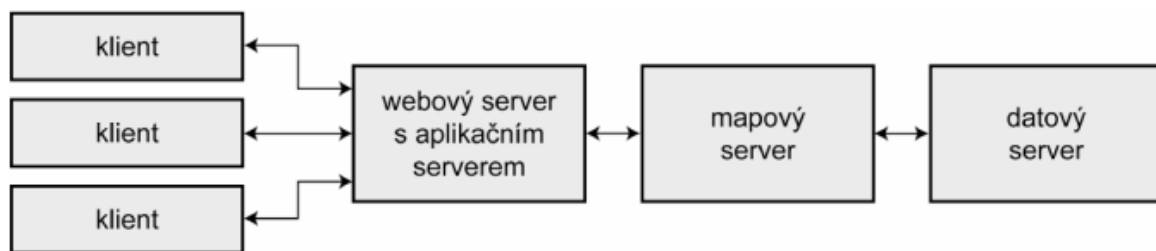
- *Internetové GIS* (Internet GIS) - využívají drátové sítě, uživatel zpravidla pracuje na stolním počítači.
- *Mobilní GIS* (Mobile GIS) - přenos dat prostřednictvím bezdrátové sítě, koncovým zařízením bývá notebook, PDA (personal digital assistant) nebo mobilní telefon.

## 2.2. Součásti internetového GIS

Není-li uvedeno jinak, v této kapitole bylo čerpáno a citováno z [Pen2003] a [Kol2004].

Řešení internetových GIS obecně obsahuje tyto čtyři hlavní součásti:

- klient,
- webový server s aplikačním serverem,
- mapový server,
- datový server.



Obrázek 2.1. Architektura obecného internetového GIS

### 2.2.1. Klient

Klient je rozhraním mezi GIS aplikací a uživatelem. Pomocí klienta uživatel ovládá aplikaci a na stejném místě je mu potom zobrazován výstup. Internetové GIS většinou jako klienta používají webový prohlížeč, případně jeho doplňky a rozšíření. Nejjednodušším příkladem klienta internetového GIS je HTML stránka s formulářem, která zobrazuje mapu jako rastrový obrázek. Uživatel aplikaci ovládá pomocí prvků tohoto formuláře, jako jsou např. přepínače, zatržítka, textová pole, obrázek mapy a potvrzovací tlačítko (různé typy elementu `<input>`, více v [Kol2004]). Takový klient ovšem nenabízí příliš velké možnosti interakce s aplikací a stránka musí být po každém odeslaném požadavku uživatele znovu načtena. Proto byla vyvinuta různá rozšíření a doplňky.

- DHTML (dynamic HTML) - sada nástrojů, díky kterým získávají statické HTML stránky dynamičtější ráz. Poté, co je stránka kompletně načtena, reaguje na některé akce uživatele, a to ihned, bez nového načítání. Například při nájezdu myši nad určitý prvek stránky se mění jeho barva, objeví se rozbalovací menu nebo okno s dalšími informacemi. Toho je dosaženo pomocí následujících technologií:

- Skriptování na straně klienta - do HTML kódu je vložena část kódu v jiném jazyce, nejčastěji se jedná o JavaScript.
- Document Object Model (DOM) - objektový model dokumentu - objektová reprezentace HTML nebo XML dokumentu, aplikační rozhraní umožňující přístup či modifikaci obsahu, struktury nebo stylu dokumentu, například právě pomocí skriptu.
- Kaskádové styly (CSS) - definují vlastnosti zobrazení pro různé prvky HTML stránky, a to hromadně a na jednom místě, díky čemuž lze pak tyto vlastnosti pomocí skriptů jednoduše měnit.

Výhodou DHTML je, že je podporováno naprostou většinou dnešních prohlížečů bez nutnosti instalace nějakého doplňku, na druhou stranu ale nenabízí příliš mnoho vylepšení funkčnosti aplikace, víceméně jen animuje stránku.

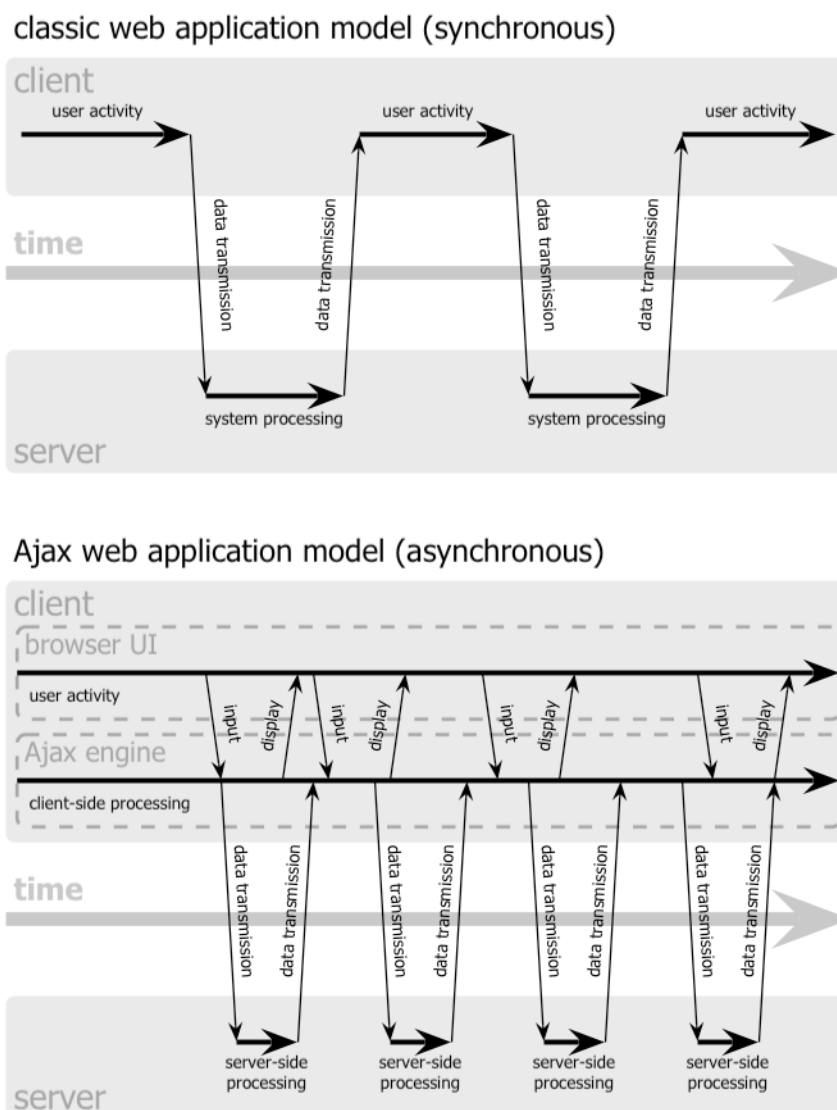
- Plug-in - „zásuvný“ programový modul, který běží nad prohlížečem a rozšiřuje jeho možnosti. Plug-in klient například uživateli umožňuje přímo pracovat s geodaty a mapami ve vektorových nebo rastrových formátech, dotazovat se na prvky mapy a provádět analýzy. Plug-in komunikuje se serverem a na vyžádání z něho nahrává data, ale tvorba mapy včetně části aplikační logiky může být umístěna na straně klienta. Nevýhodou tohoto řešení je, že každý prohlížeč vyžaduje speciální plug-in - neexistuje přenositelnost mezi různými prohlížeči.
- Java applet - spustitelný kód psaný v jazyce Java, který je stažen společně s webovou stránkou a následně vykonán na straně klienta. Jeho úloha je stejná jako v případě plug-in, tedy přenáší část aplikační logiky na stranu klienta. V případě, že se přenáší hotové rastrové mapy, Java applet jen zpříjemňuje jejich prohlížení, např. umožňuje interaktivní posun mapy. Ovšem v případě práce s vektorovými daty jsou možnosti širší, Java applet se na straně klienta stará např. o renderování mapy, zpracování prostorových dotazů a analýzy. Výhodou řešení pomocí Java applet oproti plug-in je nezávislost na operačním systému a prohlížeči, jedinou podmínkou je nutnost nainstalované podpory Javy na klientském počítači (Java Virtual Machine).
- ActiveX controls - rozšíření vytvořené firmou Microsoft, postavená na systému OLE (Object Linking and Embedding). Funkcionalita je velmi podobná Java applet řešení, tedy rozšíření možností klienta. Také princip je stejný - kód je vložen do HTML a následně vykonán na

straně klienta. Navíc ale ActiveX controls umožňují uživateli přistupovat k lokálně uloženým datům a kombinovat je s daty ze serveru. Narozdíl od Java appletu, který běží uvnitř Java Virtual Machine, ActiveX běží přímo v operačním systému a může tedy plně využívat všechny jeho prostředky, funkcionalitu, paměť a lokální soubory. Díky tomu je toto řešení výkonnější než Java applet, ale má i svoje nevýhody. Za prvé, prohlížeč musí podporovat OLE standard, což splňuje pouze Internet Explorer. Za druhé, díky možnosti přístupu ke zdrojům klientského počítače jsou ActiveX controls snadno zneužitelné - je možno napsat zákeřný ActiveX kód, který bude škodit klientskému počítači.

- Ajax - souhrnný název pro sadu technologií, které umožňují vytvářet interaktivní webové aplikace, tvářící se více jako desktopové - mění svůj obsah bez nutnosti znovunačítání celé stránky. Název Ajax je zkratkou pro „Asynchronous JavaScript and XML“, což vyjadřuje, jaké jsou základní stavební kameny této technologie. Ajax se tedy skládá z těchto komponent [i-Gar2005]:

- XHTML (nebo HTML) + CSS pro prezentaci informací,
- Document Object Model (DOM) pro dynamické zobrazování prezentovaných informací,
- XML a XSLT pro výměnu dat,
- XMLHttpRequest pro asynchronní výměnu dat s webovým serverem,
- a JavaScript pro svázání všech těchto technologií dohromady.

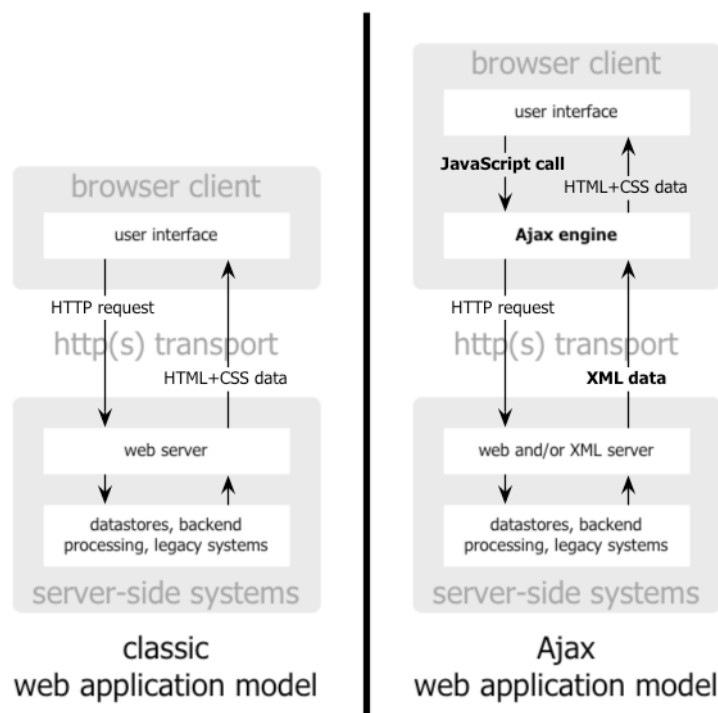
Klasická webová aplikace pracuje tak, že většina uživatelových akcí vyvolává požadavek na webový server, který po chvíli práce vrací HTML stránku. Zatímco server pracuje, uživatel musí čekat. Tuto povahu výměny informací na webu eliminuje právě Ajax, zařazením tzv. *Ajax engine* mezi klienta a server. Ten je zodpovědný jak za interpretaci výsledků uživateli, tak komunikaci se serverem. Díky tomu je výměna informací mezi uživatelem a aplikací asynchronní - nezávislá na komunikaci se serverem:



**Obrázek 2.2. Výměna informací v klasickém a Ajax webovém modelu (převzato z [i-Gar2005])**

Každá akce uživatele, která by v klasickém webovém modelu vyvolala HTTP požadavek, je v tomto případě JavaScript volání do Ajax engine. Jestliže akce uživatele nevyžaduje přenos dat ze serveru, Ajax engine obslouží požadavek bez nutnosti komunikace se serverem. V případě, že Ajax engine potřebuje data ze serveru, vysílá tyto požadavky asynchronně, bez pozastavení uživatelské interakce s aplikací.

Rozdíly mezi klasickým webovým modelem a modelem s použitím Ajax ilustruje obrázek:



**Obrázek 2.3. Architektura klasického a Ajax webového modelu (převzato z [i-Gar2005])**

Jako každá technologie, má i Ajax své výhody a nevýhody [i-Sni2005]:

- Výhody:
  - Ajax především urychluje uživateli práci, v tom je jeho hlavní výhoda. Nemusí se po každé znovu nahrávat nová stránka. Toto chování je daleko blíže tomu, co zná uživatel z klasických desktopových aplikací.
  - Dále Ajax šetří datové přenosy - u klasické webové aplikace se s každým požadavkem musí uživateli posílat celý kód stránky, v němž je většinou málo nového a důležitého. Pomocí Ajax se posílá jenom to důležité.
- Nevýhody:
  - Ajax znemožňuje použití tlačítka Zpět v prohlížeči (protože to se používá jen pro statické stránky). Uživatelé jsou na tlačítko Zpět zvyklí a očekávají od něj určitou funkci. Ajax jim ale v lepším případě vůbec neumožní ho použít, v tom horším použít půjde, ale jeho chování bude naprosto neočekávané – vrátí uživatele na předcházející stránku, nevrátí aplikaci do předcházejícího stavu (a pravděpodobně se jeho stisknutím ztratí uživateli

práce, kterou na stránce udělal). Tento problém je možné částečně řešit (např. použitím neviditelného XHTML prvku `IFRAME`), ale realizace je poněkud složitější.

- Při změnách na stránce pomocí Ajax se nemění URL v adresním řádku prohlížeče. Proto není možné takto modifikovanou stránku poslat e-mailem nebo uložit do záložek. Tento problém se dá řešit například tak, že se za URL dosazují identifikátory začínající na # (odkaz dovnitř stránky). Při opětovném vyvolání takového URL ho JavaScript zjistí a uvede stránku do příslušného stavu (tím se dá vyřešit i problém s tlačítkem Zpět).
- Problémem také může být síťová latence (zpoždění): potřeba komunikace přes Internet má negativní dopady na rychlost odezvy a interaktivitu uživatelského rozhraní. Pokud aplikace zpracovává jeho požadavek (a na pozadí komunikuje se serverem), jediné, co uživatel zaregistruje, je zpožděná reakce, a proto se může snažit operaci spustit znovu, neboť se domnívá, že systém jeho příkaz ignoroval. Z toho důvodu je vhodné uživateli nějak jasně signalizovat, kdy aplikace pracuje.
- Jako nevýhoda může být též vnímána nutnost použití moderních webových prohlížečů, které podporují potřebné technologie. Toto už ale dnes nepředstavuje velký problém, neboť všechny běžně používané prohlížeče tyto technologie alespoň v základu podporují.

### 2.2.2. Webový a aplikační server

Druhým článkem internetového GIS je webový server s aplikačním serverem. Webový server je program, který nepřetržitě běží na serverovém stroji a čeká na požadavky webových klientů. Jakmile webový server obdrží požadavek, vyhodnotí jej a podle toho, o jaký požadavek se jedná, dále reaguje jedním z následujících způsobů:

- zašle vyžádaný existující HTML dokument (příp. jen rastrový obrázek mapy) zpět klientovi,
- zašle klientovi vyžádané Java applety nebo ActiveX controls,
- předá požadavek dále ke zpracování jinému programu (např. mapovému serveru).

V případě, že webový server předává požadavek jinému programu, využívá služeb aplikačního serveru - ten je ve většině případů implementován jako modul webového serveru, případně jsou obě funkce zahrnuty v jednom programu. Aplikační server je prostředníkem mezi webovým a mapovým serverem, jehož hlavní funkcí je navazování, udržování a uzavírání spojení mezi nimi, dále interpretace požadavků klienta, vyrovnávání zátěže mezi mapovými a datovými servery (pokud je jich více) a řízení stavů, transakcí a bezpečnosti. Takovýto prostředník se nazývá

*middleware* nebo *glue*, ale protože se *middleware* vyskytuje na více místech řešení internetového GIS, v tomto případě budeme užívat označení *web middleware*. Aplikačním serverem může být např. CGI program nebo přímý modul pro webový server (SAPI), Java řešení používají tzv. *servlet engine* (nazývaný také *web container*). ESRI ArcIMS obsahuje navíc ještě jeden, vlastní aplikační server (ArcIMS Application Server), který je samostatným programem, nezávislým na webovém serveru [ESRI].

### 2.2.3. Mapový server

Mapový server je základním kamenem internetového GIS. Přes webový server přijímá a plní požadavky od klienta - vyhodnocuje prostorové dotazy, získává data, sestavuje mapu a předává ji směrem ke klientu - většinou jako hotový obrázek v rastrovém formátu (PNG, GIF, JPEG). V případě, že tvorbu mapy má na starosti klient, posílá vybrané prvky mapy.

Mapových serverů může v rámci jednoho řešení existovat více, mohou být umístěny na různých strojích a mít rozdělené úlohy tak, že každý poskytuje specifické služby. Některá řešení používají pro mapový server jiný název, ale jedná se o stejnou komponentu se stejnými funkcemi. Příkladem může být ESRI ArcIMS, kde se pro mapový server používá označení *spatial server*.

### 2.2.4. Datový server

Data pro internetový GIS mohou být uložena v souborech nebo databázích. Uložení dat v souborech se používá jen u malých projektů, větší projekty většinou spoléhají na databáze, z důvodu mnoha výhod, které databázové systémy nabízejí. Komponentou, která zajišťuje ukládání, uchovávání a přístup ke geodatům v databázové struktuře, je datový server. Ostatní součásti, jako mapový server nebo přímo webový klient, přistupují k databázi přes *SQL (Structured Query Language - strukturovaný dotazovací jazyk)*, proto se také o datovém serveru mluví někdy jako o SQL serveru. SQL je mezinárodní standard, avšak ne ve všech databázových systémech jsou implementovány všechny požadavky normy a naopak, jsou obsaženy i prvky a konstrukce, které ve standardu nejsou. Vznikají tak různé verze SQL pro různé databáze, což je důvodem, proč se často pro usnadnění přístupu k různým databázovým systémům používá *databázový middleware*. Účelem databázového *middleware* je vytvořit standardní SQL rozhraní, které si poradí s rozdíly mezi jednotlivými systémy pomocí různých databázových ovladačů. Nejznámějším databázovým *middleware* je např. ODBC či JDBC.

## 2.3. Interoperabilita v GIS podle OGC

Současný trend přechodu od desktopových GIS systémů k distribuovaným GIS službám s sebou nese problém interoperability mezi jednotlivými produkty. *Interoperabilita* je schopnost tech-

nického zařízení či software od různých výrobců spolu úspěšně komunikovat a spolupracovat [i-TS]. Chápání interoperability přesahuje schopnost integrace nesourodých dat různých datových formátů, jde i o integraci na úrovni programových aplikací, *webových* i jiných *služeb* [i-Skl2006].

*Webové služby* (*web services, WS*) je souhrnné označení pro sadu technologií umožňujících komunikaci mezi aplikacemi přes internet. Jsou postaveny na jednoduchých technologiích jako XML a HTTP a tudíž nezávislé na použité platformě [i-Kos2006].

### 2.3.1. Open Geospatial Consortium

Zachování interoperability v GIS zajišťuje vývoj standardů a specifikací a jejich používání. Jde jednak o standardizování datových formátů a struktur, ale také o standardy a specifikace pro definice výpočetních postupů, algoritmů, specifikace aplikačních rozhraní, protokolů a webových služeb. Zásadní vliv na specifikace a standardizaci v GIS má několik konsorcií. Například konsorcium W3C (World Wide Web consortium) se sice přímo standardizací v Geoinformatice nezabývá, ale má pro tento obor velký význam, neboť má zásadní vliv na vývoj interoperability ve webových technologiích vůbec. Mezi nejdůležitější subjekty, které se zabývají přímo standardizací v Geoinformatice, patří ISO (International Organization for Standardization), INSPIRE (The INfrastructure for SPatial InfoRmation in Europe) a snad nejzásadnější vliv má *Open Geospatial Consortium, Inc.*

*Open Geospatial Consortium, Inc. (OGC)* je mezinárodní průmyslové neziskové konsorcium více než 300 obchodních společností, univerzit a vládních organizací, které společně usilují o interoperabilitu v oblasti Geografických informačních systémů.

### 2.3.2. Webové služby specifikované OGC

Implementace a dodržování standardů webových služeb umožňuje úspěšnou komunikaci mezi různými servery a díky tomu i vzájemné sdílení geodat. Mapový server pak může například jako podkladové vrstvy používat také sdílené datové vrstvy ze vzdálených serverů, aniž by klient poznal, na kterém serveru jsou data fyzicky uložena. Webovou službu může také využívat silný klient, který pak se sdílenými daty pracuje stejně jako s daty uloženými na lokálním disku. Mezi nejznámější a nejčastěji využívané OGC specifikace patří specifikace následujících webových služeb:

- *Web Map Service (WMS)* je služba, která automaticky generuje vyžádanou mapu jako obraz a ten posílá klientovi. Formát tohoto obrazu je nejčastěji rastrový (PNG, GIF, JPEG), ale může být i vektorový (SVG). Podstatné ovšem je, že klient nedostává geodata jako taková. Služba funguje tak, že klient vyšle na server požadavek, ve kterém předává parametry pro



tvorbu požadovaného obrazu (formát, kartografické zobrazení atd.), podle nichž server obraz vygeneruje a pošle zpět klientovi. WMS definuje tři základní operace:

- GetCapabilities - vrací seznam možných operací a podporovaných parametrů mapového serveru
- GetMap - vrací vlastní mapu
- GetFeatureInfo - vrací popis prvku mapy (atributy)

Tyto operace mohou být vyvolány pomocí standardního webového prohlížeče, odesláním dotazu pomocí URL.

- *Web Feature Service (WFS)* ve své nejjednodušší podobě funguje velmi podobně jako WMS, ale s tím rozdílem, že nepracuje s mapou (obrazem), ale samotnými geodaty, která převádí do formátu GML (pokud v něm již nejsou) a ta potom posílá klientovi. Tato varianta se nazývá základní (basic) nebo také read-only WFS. Pomocí WFS je však také možné data na serveru modifikovat, a to díky definovaným operacím CREATE (vytváření nového prvku), UPDATE (editace) a DELETE (mazání). Takovou implementaci nazýváme transakční (transaction) WFS. Klient a server spolu komunikují přes protokol HTTP.

Další OGC webové služby:

- WCS (Web Coverage Service),
- WTS (Web Terrain Service),
- WPS (Web Processing Service) - navrhovaná specifikace.

## Kapitola 3

# Popis struktury výměnného formátu informačního systému katastru nemovitostí České republiky

*Výměnný formát informačního systému katastru nemovitostí České republiky (VFK)* je přesně definovaná struktura pro předávání dat mezi *informačním systémem katastru nemovitostí (ISKN)*, spravovaným *Českým úřadem zeměměřickým a katastrálním (ČÚZK)* a jinými informačními systémy. Soubor výměnného formátu (VFK) je textový soubor s kódováním češtiny dle ISO 8859-2 (ISO Latin2), ve výjimečných případech WIN1250, případně soubor ve formátu XML verze 1.0 s kódováním češtiny dle WIN1250. Z hlediska obsahu údajů jsou oba formáty totožné [ČÚZK2007].

### 3.1. Části VFK

Soubor VFK se skládá z těchto částí:

- hlavičky,
- datových bloků,
- koncového znaku &K.

#### 3.1.1. Hlavička

Řádek hlavičky souboru VFK začíná návěštím &H, po kterém následuje označení položky a příslušné údaje oddělené středníkem, např.:

### Příklad 3.1. Řádek hlavičky souboru VFK

```
&HJMENO;"Jméno Příjmení"
```

Hlavička obsahuje následující řádky:

- označení verze VFK,
- datum a čas vytvoření souboru,
- původ dat,
- označení kódové stránky,
- seznam skupin datových bloků souboru,
- jméno osoby, která soubor vytvořila,
- časovou podmínku použitou pro vytvoření souboru,
- omezující podmínku - katastrální území,
- omezující podmínku - oprávněné subjekty,
- omezující podmínku - parcely,
- omezující podmínku - polygon.

Způsob zápisu konkrétních řádek hlavičky je uveden v [ČÚZK2007].

#### 3.1.2. Datové bloky

Každý datový blok souboru VFK obsahuje následující řádky:

- právě jeden uvozující řádek obsahující seznam atributů s jejich datovými typy (v požadovaném pořadí),
- řádky obsahující vlastní data (ve stanoveném pořadí).

Uvozující řádek bloku začíná návěštím &B, následuje zkratka označení bloku, např. PAR (parcela) a jednotlivé atributy oddělené středníkem, např.:

### Příklad 3.2. Uvozující řádek bloku souboru VFK

```
&BPAR;ID N30;STAV_DAT N2;DATUM_VZNIKU D;DATUM_ZANIKU D;PRIZNAK_KONTEXTU N1; ...
```

Atribut obsahuje název a typ položky. Přípustné typy položek jsou:

- text (řetězec), označený TX, kde X označuje maximální délku řetězce,
- číslo, označený NX.Y, kde X vyjadřuje maximální počet míst a Y udává, kolik míst z X leží za desetinnou tečkou,
- datum, označený D, ve tvaru DD.MM.YYYY HH:MI:SS.

Řádek obsahující vlastní data začíná návěštím &D, následuje zkratka označení bloku, např. PAR (parcela) a vlastní data oddělená středníkem. Pro každý objekt je vytvořen jeden řádek. Viz příklad:

### Příklad 3.3. Řádek s daty souboru VFK

```
&DPAR;92340708;0;"09.09.1999 00:00:00";"";2; ...
```

Datové bloky jsou dle tematiky rozděleny do skupin, z nichž mohou být na přání uživatele exportovány jen ty skupiny, které potřebuje:

Název skupiny	Kód	Popis
Nemovitosti	NEMO	parcely a budovy
Jednotky	JEDN	bytové jednotky
Bonitní díly parcel	BDPA	kódy BPEJ k parcelám
Vlastnictví	VLST	listy vlastnictví, oprávněné subjekty a vlastnické vztahy
Jiné právní vztahy	JPVZ	ostatní právní vztahy kromě vlastnictví
Řízení	RIZE	údaje o řízení (vklad, záznam)
Prvky katastrální mapy	PKMP	digitální katastrální mapa
BPEJ	BPEJ	hranice BPEJ včetně kódů
Geometrický plán	GMPL	geometrické plány (v současné době z technických důvodů neposkytováno)
Rezervovaná čísla	REZE	rezervovaná parcelní čísla

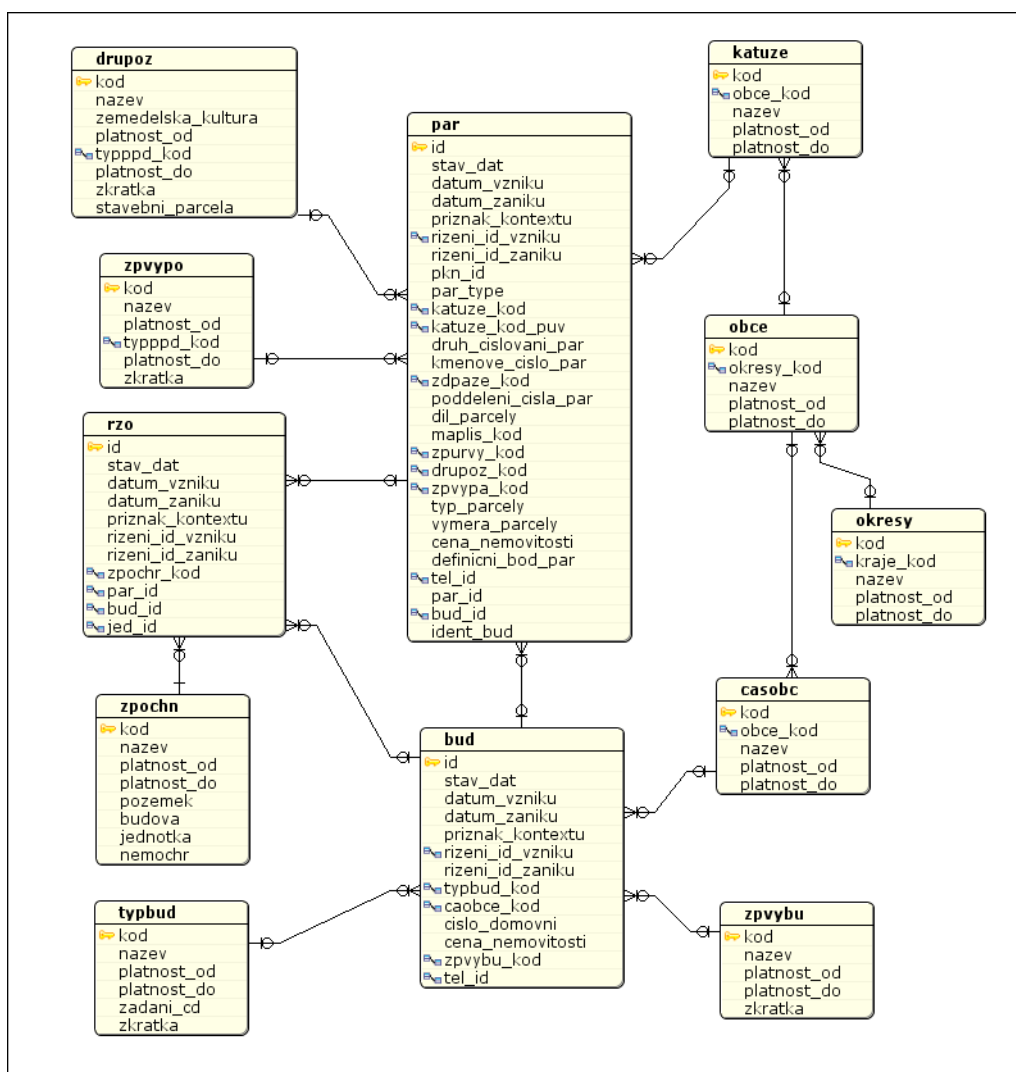
Tabulka 3.1. Skupiny datových bloků

## 3.2. Popis vybraných skupin datových bloků

Tato práce se zabývá výpisem atributových informací o parcelách a budovách, proto budou popsány především s tím související datové bloky (tabulky) a vztahy mezi nimi. Z toho důvodu také není popsán způsob uložení prostorových informací - touto problematikou se zabývá např. [Jed2006] a [Ora2006]. Do datových modelů a popisu jednotlivých skupin nejsou vždy zahrnuty všechny datové bloky, ale jen ty, které jsou z pohledu využitelnosti v práci důležité.

### 3.2.1. Nemovitosti (NEMO)

Datové bloky ve skupině nemovitosti obsahují atributové informace k nemovitostem (parcelám a budovám), jako druh pozemku, způsob využití pozemku a budovy, údaje jednoznačné identifikace nemovitostí apod. Většina informací je uložena přímo v tabulkách *PAR* a *BUD* v podobě cizích klíčů ostatních na ně navázaných tabulek, které jsou většinou číselníky. Vazby mezi vybranými tabulkami této skupiny jsou patrné z diagramu, jednotlivé tabulky jsou popsány níže.



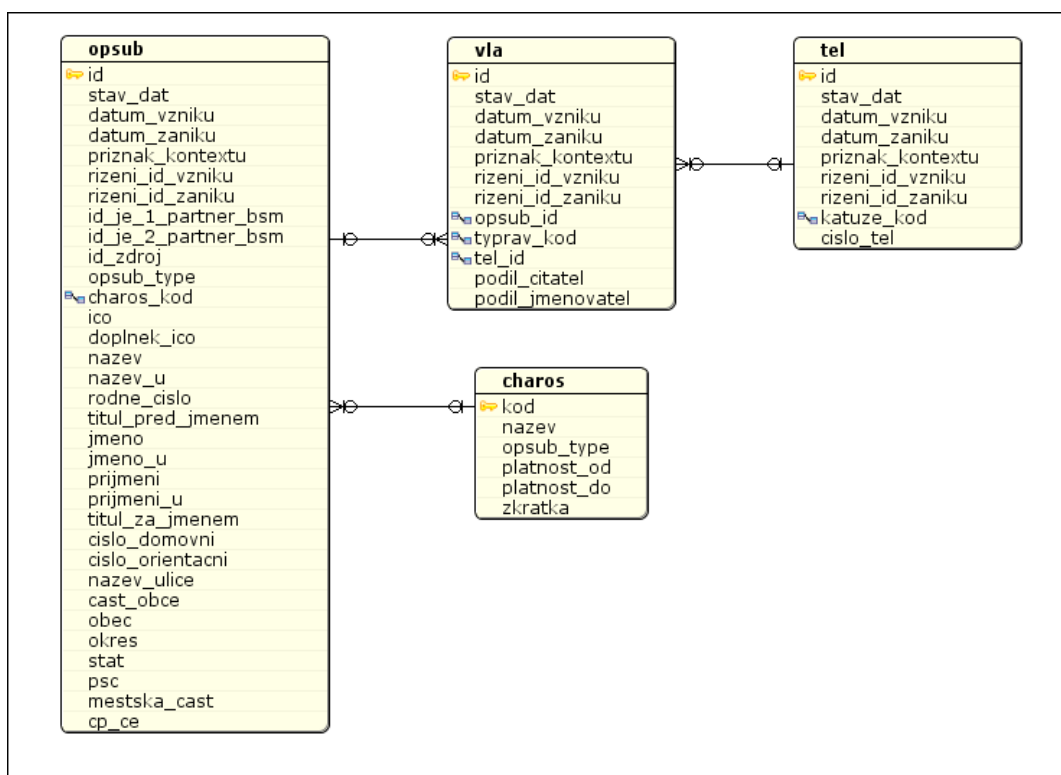
Obrázek 3.1. Datový model vybraných tabulek skupiny NEMO

- PAR (Parcely) - obsahuje atributové údaje o parcelách - informace potřebné k jednoznačné identifikaci parcely (kód katastrálního území, druh číslování parcel, kmenové číslo parcely, případně poddělení čísla parcely a číslo dílu parcely), dále druh pozemku a způsob využití, výměra parcely, identifikátor listu vlastnictví, na kterém se parcela nachází a identifikátor budovy ležící na parcele. Na jedné parcele se může nacházet pouze jedna budova, ale jedna budova může ležet na více parcelách - proto je mezi entitami PAR a BUD vazba N:1.
- BUD (Budovy) - zde se nachází atributové informace o budovách - jednoznačná identifikace budovy (kód části obce, typ budovy, domovní číslo), kód způsobu využití budovy a identifikátor listu vlastnictví. Budova na cizím pozemku má jiné TEL\_ID než parcela, na které leží.
- DRUPOZ (Druh pozemku) - číselník druhů pozemků.

- ZPVYPO (Způsob využití pozemku) - číselník definující pro parcelu druh způsobu využití pozemku v rámci druhu pozemku.
- TYPBUD - číselník obsahující typy budov (budova s číslem popisným, číslem evidenčním...).
- KATUZE (Katastrální území) - číselník katastrálních území.
- OBCE - číselník všech obcí v ČR.
- CASOBC (Části obcí) - číselník částí obcí.
- OKRESY - číselník okresů ČR.
- ZPOCHN - číselník obsahující vybrané způsoby ochrany nemovitosti z oblasti ochrany přírody, kultury nebo zdravotnictví a umístění geodetického bodu na parcele.
- RZO - Vazební tabulka přiřazující způsob ochrany nemovitosti ke konkrétní parcele, budově nebo jednotce (realizace vazby M:N mezi entitami *ZPOCHN* a *PAR* (příp. *BUD*).
- ZPVYBU - číselník, který definuje způsob využití budovy.

### 3.2.2. Vlastnictví (VLST)

Datové bloky této skupiny definují vlastnický vztah mezi oprávněným subjektem (datový blok *OPSUB*) a listem vlastnictví (tělesem, datový blok *TEL*).



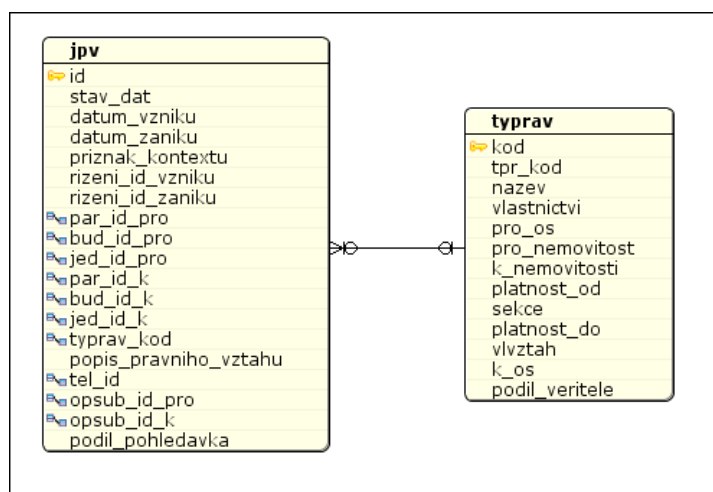
Obrázek 3.2. Datový model skupiny VLST

- OPSUB (Oprávněný subjekt) - vlastník či jiný oprávněný. Oprávněným subjektem může být fyzická či právnická osoba, manželé v bezpodílovém spoluvlastnictví, obec, stát apod.
- TEL (Tělesa) - tabulka představující listy vlastnictví.
- VLA (Vlastnictví) - tabulka realizující vlastnické vztahy – vazby oprávněných subjektů k nemovitostem zapsaným na jednotlivých listech vlastnictví. Dále definuje také typ právního vztahu a podíl (v případě více vlastníků)
- CHAROS (Charakteristika oprávněného subjektu) - číselník blíže charakterizující oprávněný subjekt.

### 3.2.3. Jiné právní vztahy (JPVZ)

Jiný právní vztah vyjadřuje jiný než vlastnický vztah jednoho subjektu (oprávněného subjektu nebo nemovitosti) k jednomu předmětu (k jedné nemovitosti, k vlastnictví nebo dalšímu jinému právnímu vztahu).



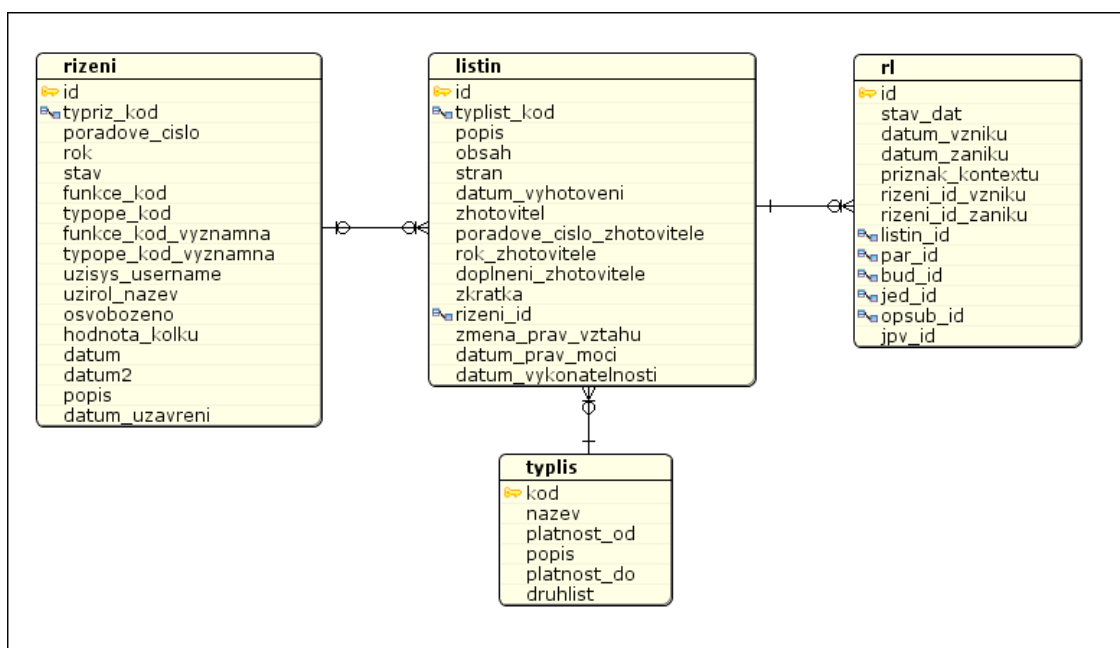


Obrázek 3.3. Datový model skupiny JPVZ

- JPV (Jiné právní vztahy) - tabulka realizující a blíže charakterizující jiné právní vztahy. Vazby  $*_K (PAR\_ID\_K, BUD\_ID\_K, JED\_ID\_K)$  definují nemovitosti či OS, které jsou „povinné“, vazby  $*\_PRO (PAR\_ID\_PRO, BUD\_ID\_PRO, JED\_ID\_PRO)$  definují „oprávněné“ nemovitosti či OS.
- TYPRAV (Typ právního vztahu) – číselník blíže charakterizující typ právního vztahu, zahrnuje např. vlastnictví, věcné břemeno, právo hospodaření s majetkem státu, právo trvalého užívání apod. Mimo jiné obsahuje i informaci o tom, v jaké sekci Výpisu z katastru nemovitostí má být tento typ právního vztahu zařazen.

### 3.2.4. Řízení (RIZE)

Tato skupina obsahuje informace o řízeních v katastru nemovitostí (např. o vkladu, záznamu, pozemkových úpravách, obnově operátu apod.). Vazby mezi vybranými datovými bloky skupiny znázorňuje diagram:

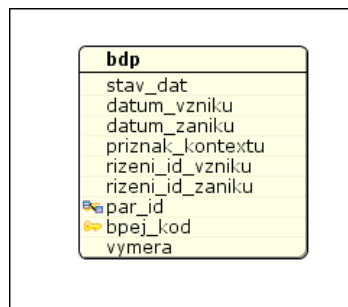


**Obrázek 3.4. Datový model vybraných tabulek skupiny RIZE**

- RIZENI (Řízení) - tabulka obsahuje základní atributy řízení (typ řízení, pořadové číslo řízení, data zahájení a ukončení atd.).
- LISTIN (Listiny) - obsahuje základní údaje charakterizující listinu jako podklad rozhodování a/nebo zápis do KN v rámci řízení. Obsahuje informace jako popis listiny, jméno a pořadové číslo zhotovitele listiny, rok vyhotovení atd.
- TYPLIS (Typ listiny) - číselník typu listin pro zařazení listin do kategorií dle použití, významu a právních úkonů spojených s listinami. Jedná se například o typy: smlouva o převodu vlastnictví nemovitosti, smlouva o zástavním nebo podzástavním právu, smlouva o oprávnění, které odpovídá věcnému břemenu, dohoda o vydání věci apod.
- RL - vazební tabulka mezi nemovitostmi a vlastnictvími na nabývací tituly. V tabulce *RL* vždy platí, že pokud je vyplněn údaj *OPSUB\_ID*, je také vyplněný údaj *PAR\_ID* nebo *BUD\_ID*.

### 3.2.5. Bonitní díly parcel (BDPA)

Ve skupině se nachází jediná tabulka:



Obrázek 3.5. Datový model skupiny BDPA

- BDP (Bonitní díly parcel) - tabulka slouží k uložení informací o bonitních dílech parcely. Obsahuje kódy BPEJ a výměry jednotlivých bonitních dílů parcely.

### 3.3. Parcelní a domovní čísla ve VF ISKN

Ve výměnném formátu ISKN nejsou uložena identifikační čísla nemovitostí vcelku, ale po částech, ve více sloupcích. Chceme-li zobrazovat atributové informace nemovitostí, je tedy nutné tato čísla rekonstruovat. Parcelní číslo se skládá ze 3 částí:

- Sloupec *DRUH\_CISLOVANI\_PAR* rozlišuje stavební parcely od pozemkových, číslo 1 značí stavební parcely, před jejichž číslo se píše "St.", před číslo pozemkových parcel, označených číslem 2, se nepíše nic.
- Ve sloupci *KMENOVE\_CISLO\_PAR* se nachází kmenové číslo parcely, tedy to, co je před lomítkem.
- Sloupec *PODDELENI\_CISLA\_PAR* je nepovinný, v případě, že je vyplněn, má parcela více poddělení a číslo nacházející se v tomto sloupci se píše za lomítko.

Domovní čísla se skládají ze dvou částí:

- Ve sloupci *TYPBUD\_KOD* tabulky *BUD* je uložen číselný kód typu budovy: 1 - budova s číslem popisným, 2 - budova s číslem evidenčním, 3 - budova bez čísla popisného nebo evidenčního, 4 - rozestavěná budova, 5 - poschodová garáž. Pro každý typ se používá zkratka, která se nachází ve sloupci *ZKRATKA* tabulky *TYPBUD*. Chceme-li tedy získat tuto zkratku, je třeba v SQL dotazu tabulky *BUD* a *TYPBUD* propojit vazbou.
- Číslo budovy se nachází ve sloupci *CISLO\_DOMOVNI* a zadává se jen u budov typu 1 a 2.

## Kapitola 4

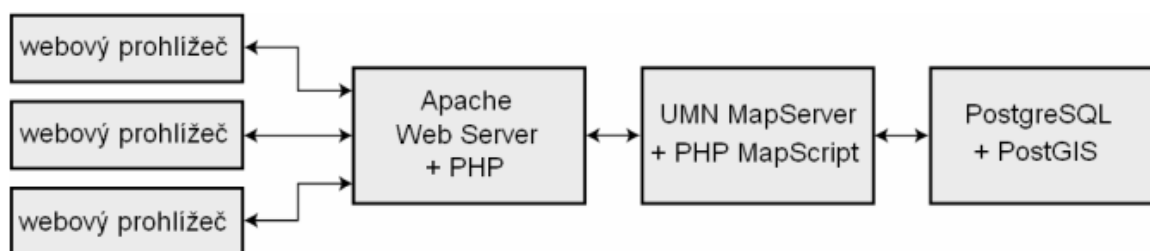
# Implementace zvoleného mapového serveru pro katastrální data

Hlavním cílem této práce bylo implementovat mapový server pro vizualizaci prostorových, ale především atributových dat katastru nemovitostí. Výchozím bodem pro tento úkol byly závěry a praktické výsledky diplomové práce J. Orálka (viz [Ora2006]). Ten zde navrhl řešení mapového serveru, postaveného na *Open Source* technologiích, a dále vytvořil skript pro import dat z výměnného formátu ISKN do databáze PostGIS.

Za *Open Source* se pokládají takové aplikace, které jsou šířeny se zachováním určitých práv a svobod pro jejich koncového uživatele (tedy nabyvatele licence). Jde o práva spouštět program za jakýmkoliv účelem, studovat, jak program pracuje a přizpůsobit ho svým potřebám (předpokladem k tomu je přístup ke zdrojovému kódu), redistribuovat kopie dle svobodné vůle, vylepšovat program a zveřejňovat tato zlepšení [i-OSS].

### 4.1. Konkrétní řešení mapového serveru

Zvolené řešení tedy vychází z návrhu prezentovaného v [Ora2006]. Kritériem pro výběr jednotlivých technologií bylo, aby spadaly do kategorie Open Source, což všechny zvolené produkty splňují. Navíc se jedná o programy zavedené, prověřené velkým počtem uživatelů, které dosahují kvalit programů komerčních. Vysoký počet uživatelů přináší také nezanedbatelnou výhodu dostatečné dokumentace a uživatelské podpory na internetu.



**Obrázek 4.1. Architektura zvoleného řešení mapového serveru**

- PostgreSQL + PostGIS - databáze pro mapový server musí podporovat ukládání prostorových dat. Standardní relační databázi PostgreSQL s nadstavbou PostGIS pro prostorová data je zřejmě tím nejlepším, co lze v kategorii Open Source najít. PostGIS obsahuje mnoho funkcí sloužících k usnadnění práce s prostorovými daty a podporuje prostorové datové typy definované OGC: point, linestring, polygon, multipoint, multilinestring, multipolygon a geometry-collection [Ora2006]. Pro obecné informace o datových serverech viz 2.2.4 – „Datový server“.
- UMN MapServer - vývojové prostředí pro vytváření webových mapových aplikací. Nejedná se o plnohodnotný GIS, ale umožňuje publikaci a sdílení geodat na internetu. Je postaven na řadě populárních nástrojů jako GDAL, OGR nebo Proj.4 a podporuje OGC standardy WMS a WFS. O jeho kvalitách svědčí fakt, že je používán mnoha komerčními i nekomerčními projekty jako základní kámen různých mapových aplikací [Kol2004]. Více o funkci mapového serveru viz 2.2.3 – „Mapový server“.
- Apache Web Server - Open Source webový (HTTP) server. Kvalitní program, na kterém běží přes 70% stránek na Internetu [i-Apa]. Pro obecné informace o webových serverech viz 2.2.2 – „Webový a aplikační server“.
- PHP - velmi rozšířený skriptovací jazyk, který je zvláště vhodný pro tvorbu webových aplikací, neboť lze začlenit přímo do HTML kódu. Kód se vykoná na straně serveru a zpět klientovi se posílá pouze HTML, takže zdrojový kód zůstává uživateli utajen. Velkou výhodou PHP je také přímá podpora velkého množství databází pomocí vestavěných funkcí. V našem konkrétním řešení mapového serveru funguje PHP jako aplikační server - prostředník mezi webovým a mapovým serverem. Více o funkci aplikačního serveru viz 2.2.2 – „Webový a aplikační server“.

## 4.2. Instalace PostGIS a import dat VF ISKN

První komponentou, kterou je nutno mít pro spuštění mapového serveru, je SŘBD PostgreSQL s nadstavbou PostGIS. Existuje v různých verzích pro různé operační systémy, stačí jen vybrat a stáhnout správný soubor. PostgreSQL nalezneme na adrese <http://www.postgresql.org/download/>, PostGIS na <http://postgis.refractory.net/download/>. Kromě toho, abychom vybrali správnou verzi pro náš operační systém, musíme také dát pozor, abychom stáhli kompatibilní verze PostgreSQL a PostGIS. Vybereme-li si například PostgreSQL verze 8.2, musíme stáhnout nadstavbu PostGIS pro tuto verzi (soubory jsou zřetelně popsány).

Začínáme instalací PostgreSQL. Postup instalace se liší v závislosti na operačním systému, v případě OS Windows lze využít instalátor, s jehož využitím je instalace poměrně snadná, návod lze nalézt např. na <http://pginstaller.projects.postgresql.org/>. Zmíním i český návod na adrese <http://gis.zcu.cz/studium/ugi/referaty/05/PostGIS/>, který jsme psali v rámci semestrální práce, a je nutné brát ho poněkud s rezervou. Již během instalace PostgreSQL je nabídnuta možnost instalovat v jednom kroku i PostGIS, což se ale v případě, že chceme nejnovější verzi, obecně nedoporučuje.

Jestliže máme PostgreSQL úspěšně nainstalován, můžeme přistoupit k instalaci rozšíření PostGIS. Ta je v případě OS Windows řešena opět pomocí instalátoru, který nás provede všemi kroky. Po úspěšné instalaci je třeba založit novou databázi pro data VF ISKN - to je možné provést buď pomocí příkazové řádky, nebo využitím nástroje PgAdmin III, což je grafické uživatelské rozhraní pro správu databáze PostgreSQL. Založení prostorové databáze je jednoduché - při instalaci PostGIS byla vytvořena databáze `template_postgis`, kterou využijeme jako šablonu (`template`). Zvolíme jméno nové databáze, například `'nvf'`. V PgAdmin III existuje pro vytvoření databáze formulář, v SQL bude příkaz vypadat např. takto:

### Příklad 4.1. SQL příkaz pro vytvoření databáze

```
CREATE DATABASE nvf
WITH OWNER=root
ENCODING='UTF-8'
TEMPLATE=template_postgis
```

Máme-li vytvořenou prázdnou databázi, můžeme přikročit k importu. K tomu lze využít skriptu, který napsal J. Orálek v rámci své diplomové práce. Vzhledem k tomu, že je napsán ve skriptovacím jazyce Python, bude nejprve nutné nainstalovat interpret tohoto jazyka, který nalezneme na stránce <http://www.python.org/download/>. Instalace na OS Windows je opět jednoduchá, pro tento OS se také může hodit rozšíření `pywin32`, které lze stáhnout na adrese <http://sourceforge.net/projects/pywin32>. Toto rozšíření obsahuje GUI PythonWin, které lze

následně použit pro spuštění importu. Dále je nutné mít správně nastaven ovladač ODBC, kterého skript využívá pro propojení s databází. Toto nastavení nalezneme v operačním systému Windows XP v **Ovládací panely/Nástroje pro správu/Datové zdroje (ODBC)**. Pro více informací o importu VF ISKN do databáze PostGIS viz [Ora2006].

Funkcí tohoto skriptu není jen import dat ze souboru VFK a vytvoření vazeb mezi tabulkami pomocí primárních a cizích klíčů, ale hlavně vytvoření geometrie u některých tabulek. Ta je uložena ve sloupci typu *GEOMETRYCOLUMN*. Pro nás je nejdůležitější geometrie parcel a budov, která je uložena ve sloupci *GEOMPAR* tabulky *PAR*, resp. *GEOMBUD* v tabulce *BUD*. Kvůli snadnější vizualizaci v silných klientech byly navíc do tabulky *PAR* „nakopírovány“ druhy pozemků a způsoby využití pozemků - tabulka se tím rozrostla o sloupce *DRUPOZ\_NAZEVA* a *ZPVYPO\_NAZEVA*.

Nově byly do tohoto skriptu také doprogramovány dvě funkce, které přidávají k tabulkám *PAR* a *BUD* nový sloupec s kompletním parcelním, resp. domovním číslem. To umožní zobrazení kompletních parcelních a domovních čísel v mapě. MapServer umí popisky (labels) vytvářet pouze z jediného sloupce tabulky, ale tyto údaje jsou ve výměnném formátu ISKN uloženy po částech (viz 3.3 – „Parcelní a domovní čísla ve VF ISKN“).

O tom, zda byl import úspěšný, se můžeme přesvědčit např. v nástroji PgAdmin III, kde by se v naší databázi nyní měly nacházet tabulky ISKN. Pro prohlížení dat lze využít aplikace JUMP, resp. OpenJUMP (<http://www.openjump.org>) nebo Quantum GIS (<http://qgis.org>), které umožňují vizualizaci geodat uložených v databázi PostGIS.

### 4.3. Instalace mapového a webového serveru

Následuje instalace UMN MapServer s PHP MapScript, webového serveru Apache a PHP jako aplikačního serveru, který bude zajišťovat jejich vzájemnou komunikaci. U všech těchto produktů opět platí, že je možné je stáhnout a nainstalovat na různé operační systémy. Pro OS Windows existuje balík MapServer for Windows (MS4W), který kromě mapového serveru a dalších souvisejících nástrojů obsahuje i Apache a PHP, takže nám výrazně ulehčí práci. Nalezneme jej na adrese <http://www.maptools.org/ms4w/>. Nejedná se o instalátor v pravém slova smyslu, ale o ZIP archív, který je nutno rozbalit do kořenového adresáře disku. Spuštěním souboru `apache-install.bat` se nainstaluje a spustí Apache jako služba. O tom, že instalace MS4W funguje, je možné se přesvědčit otevřením webového prohlížeče a zadáním adresy `http://localhost`. Měla by se zobrazit úvodní stránka MS4W, na které lze nalézt informace o dalších komponentách, které jsou součástí balíku.

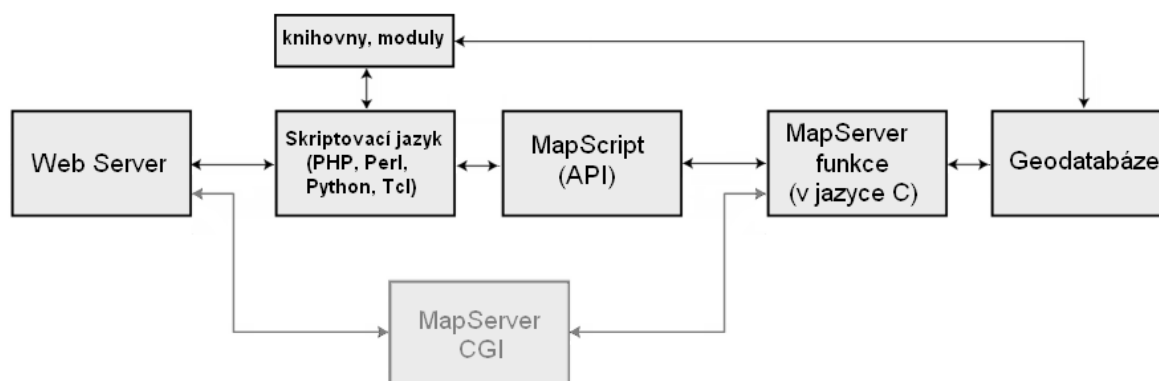
V případě, že chceme Apache a PHP instalovat samostatně (díky rozsáhlé uživatelské podpoře to není velký problém), nebo je již máme na počítači instalované, existuje možnost na adrese [http://www.maptools.org/php\\_mapscript/](http://www.maptools.org/php_mapscript/) stáhnout samotný PHP MapScript a MapServer.

V tom případě je nutné jednotlivé komponenty vzájemně nakonfigurovat a také dát pozor, abychom stáhli správnou verzi PHP MapScript, kompatibilní s nainstalovaným PHP.

### 4.3.1. MapScript vs. CGI

Představme si MapServer jako soubor funkcí napsaných v jazyce C. K této funkcionalitě je možné přistupovat dvěma způsoby. Buď pomocí CGI programu `mapserv.exe`, nebo s použitím MapScript. CGI program je aplikace napsaná v C, která je spuštěna webovým serverem při každém požadavku a má přímý přístup k funkcionalitě MapServeru. Obsahuje již množství komplexních funkcí, jako je např. zoomování, posun, generování legendy nebo vypínání vrstev, což je na jednu stranu výhoda, ale také představuje určité omezení. Naproti tomu MapScript je pouze API, které umožňuje přístup k jednotlivým třídám a metodám MapServeru pomocí skriptovacích jazyků (pro různé jazyky existuje různé API). Funkce jako zmíněné zoomování či zapínání vrstev je třeba naprogramovat, ale nabízí se mnohem širší možnosti, jak to provést. Jinými slovy, pomocí MapScriptu lze naprogramovat veškerou funkcionalitu CGI a k tomu ještě mnohem více [i-MS-Ex].

Při použití MapScriptu není využíván CGI program, stejně jako není třeba mít nainstalovaný skriptovací jazyk a MapScript, používáme-li CGI. Rozdíl mezi řešením MapServeru s MapScriptem oproti CGI ilustruje schéma:



Obrázek 4.2. Schéma MapServeru a MapScript (podle [i-MS-Ex])

## 4.4. Vizualizace prostorových dat

Publikováním geodat pomocí UMN MapServeru a jeho možnostmi se podrobněji zabývá např. [Kol2004], proto bude další popis omezen jen na toto konkrétní řešení a jeho specifika.



### 4.4.1. Mapsoubor

Mapsoubor (mapfile) je základním konfiguračním souborem MapServeru. Umožňuje nastavení parametrů všech prvků výstupu: vlastní mapy, legendy, přehledové mapky nebo grafického měřítka. Mapsoubor má příponu map a umísťuje se do adresáře, kam není povolen přístup z internetu (tedy jinam než např. html soubory).

Obsah mapsouboru má hierarchickou strukturu, skládá se z objektů, které začínají klíčovým slovem a končí slovem END. Každý objekt obsahuje definovanou množinu povinných i nepovinných parametrů a jejich hodnot, a případně dalších podobjektů. Na vrcholu této hierarchie stojí objekt MAP. Ten může dále obsahovat objekty jako REFERENCE (přehledová mapka), LEGEND (legenda), SCALEBAR (měřítko), nebo PROJECTION (použité zobrazení). Základním podobjektem objektu MAP je LAYER, kterých zpravidla mapsoubor obsahuje celou řadu. LAYER definuje vše co se týká jednotlivých vrstev, od zdroje dat po způsob zobrazení vrstvy. Podrobnou referenční příručku k mapsouboru lze najít na adrese <http://mapserver.gis.umn.edu/docs/reference/mapfile>.

Mapsoubor pro moji aplikaci má následující strukturu:

#### Příklad 4.2. Ukázka mapsouboru (nekompletní)

```
MAP
  NAME Katastr          # kazdy objekt by mel byt pojmenovan
  SIZE 600 450          # velikost mapy v pixelech
  EXTENT -654000 -1071000 -649800 -1068000 # hranicni souradnice
  UNITS METERS          # mapove jednotky
  ...
  WEB
    IMAGEPATH "C:\web\mapserver\temp\" # adresar, kam se budou ukladat obrazky
    IMAGEURL  "/mapserver/temp/"      # URL adresare s obrazky
  END

  REFERENCE # referencni mapa
    IMAGE 'C:\MapServer\mapservdata\refmap.png' # Obrazek referencni mapy
    SIZE 200 150 # velikost referencni mapy
    ...
  END

  SCALEBAR # vlastnosti grafickeho meritka
  ...
  END # Scalebar
  ...
#####
# zacatek vseh vrstev
```

```

LAYER
  NAME      parcels
  TYPE      POLYGON
  ...
  CLASS
    NAME 'Parcely'
    STYLE
      ...
    END # style
  END # class parcels
END # layer parcels

# dalsi vrstvy...

# konec vseh vrstev
#####
END # konec mapfilu

```

Tento mapsoubor se skládá z šesti vrstev:

- Parcely - vrstva zobrazující všechny parcely. Zdrojem dat je sloupec *GEOMPAR* v tabulce *PAR*, obsahující geometrii typu POLYGON pro parcely. Vrstva se dá zapínat a vypínat, implicitně je zapnutá.
- Budovy - vrstva budov, zdrojem dat je sloupec *GEOMBUD* v tabulce *BUD*, obsahující geometrii budov. Stejně jako vrstva parcel je tato vrstva implicitně zapnutá.
- Parcely\_label - vrstva parcelních čísel. Vrstva je implicitně vypnutá a dá se zapnout pouze v případě, že je zapnutá i vrstva parcel. Popisky jsou nastaveny tak, že se zobrazují až při určitém přiblížení parcely (záleží na počtu pixelů, který parcela v tom momentě zaujímá).
- Budovy\_label - obdoba vrstvy parcelních čísel, která zobrazuje domovní čísla. Pro odlišení od parcelních čísel jsou domovní čísla zobrazována kurzívou a jinou barvou.
- Parcely\_selected - vrstva, která slouží k vizualizaci vybrané parcely v mapě. Vrstva je implicitně vypnutá, pouze v případě dotazu na parcelu je na ní aplikován filtr (ze všech parcel se vybere pouze dotazovaná) a vrstva je ve skriptu zapnuta. Barva výplně je u této vrstvy nastavena na průhlednou a obrys je realizován čarou tloušťky 3 pixely. Vzhledem k tomu, že tato vrstva je umístěna "nad" vrstvou parcel, dojde tedy ke zvýraznění obrysu vybrané parcely.<sup>1</sup>

<sup>1</sup>MapServer poskytuje možnost zvýraznění vybrané parcely pomocí objektu QUERYMAP, ale pouze formou jiné výplně, ne obrysu.

- `Budovy_selected` - podobně jako vrstva `parcely_selected` slouží tato vrstva k vizualizaci vybraných budov.

#### 4.4.2. Stránka s mapou

K zobrazování mapy slouží stránka `index.php`. Jedná se PHP skript, který dynamicky generuje HTML stránku s mapou a jejími ovládacími prvky. K ovládání aplikace slouží tyto prvky:

- Přepínač nástrojů - určuje, co se má stát v případě, že uživatel klikne do mapy, konkrétně:
  - **Posun** vycentruje mapu tak, že bod kliknutí se přesune doprostřed mapy,
  - **Přiblížit** zvětší měřítko o zadaný **zoom faktor** (viz dále) a vycentruje mapu na bod kliknutí,
  - **Oddálit** také centruje mapu, ale měřítko naopak zmenší,
  - **Dotaz** slouží k výběru geoprvcu z určené vrstvy (viz dále), s mapou nijak nehýbe.
- **Zoom faktor** - určuje koeficient, kterým se násobí, resp. dělí měřítko mapy při přibližování, resp. oddalování. Implicitně nastaven na 2, hodnota se dá přepsat,
- **Full extent** - tlačítko vrací mapu na pozici a měřítko implicitně dané v mapsouboru,
- **Výběr vrstvy** - určuje do jaké vrstvy se dotazovat,
- **Zaškrťovací čtverečky** pro výběr vrstev, které mají být zobrazeny,
- **Obnovit** - poslouží v případě, že uživatel změní zatržení u vrstev a nechce hýbat s mapou. Mapa se znovu načte ve stejné pozici, ale s vrstvami zobrazenými dle požadavku.
- Ovládacím prvkem je i **přehledová mapa** - kliknutí do ní vyvolá centraci hlavní mapy na zadané místo při zachovaném přiblížení, ať je použitý nástroj jakýkoli.
- Mezi ovládací prvky lze zařadit i samotné **mapové pole**, neboť kliknutí do něj vyvolává reakci dle použitého nástroje.

Všechny ovládací prvky jsou elementy formuláře. Princip ovládání je postaven na tom, že je-li kliknuto do mapy, referenční mapy nebo zmáčknuto jedno z tlačítek, dojde k odeslání formuláře, přesněji řečeno všech hodnot parametrů formuláře, na webový server. Následující tabulka obsahuje přehled parametrů a hodnot, kterých mohou tyto parametry nabývat:

parametry	možné hodnoty	popis
tool	"pan", "zin", "zout", "query"	hodnota značí jeden ze čtyř nástrojů, který byl použit
zsize	číslo	hodnota zoom faktoru
full	"full extent"	indikuje, že bylo zmáčknuto tlačítko <b>Full extent</b>
qlayer	"parcely", "budovy"	hodnota říká, která vrstva byla vybrána pro dotazování
budovy, budovy_label, parcely, parcely_label	"displayed"	hodnota "displayed" u těchto parametrů značí, že čtvereček u příslušné vrstvy byl zaškrtnut
renew	"obnovit"	označuje, že bylo stisknuto tlačítko <b>Obnovit</b>
mapa_x, mapa_y	číslo	tyto parametry jsou odeslány v případě, že bylo kliknuto do mapy - obsahují souřadnice bodu kliknutí v pixelech
ref_x, ref_y	číslo	podobné jako předchozí případ, ale odesílají se v případě, že bylo kliknuto do referenční mapy
last_extent	čtveřice čísel	parametr ze skrytého formulářového pole - hodnoty naposled použitých hraničních souřadnic
last_scale	číslo	parametr ze skrytého formulářového pole - hodnota naposled použitého měřítka

**Tabulka 4.1. Parametry formuláře a možné hodnoty**

Na webovém serveru jsou odeslané parametry zpracovány pomocí PHP skriptu a na základě jejich hodnot je rozhodnuto, co se má dále provést. Ve všech případech jsou nakonec spočítány nové hraniční souřadnice (obdélník ohraničující mapový výřez) a poté webový server pošle mapserveru požadavek na sestavení nového obrázku mapy. Hotový obrázek je uložen do určitého adresáře, přístupného z internetu, a mapový server pošle zpět webovému serveru URL adresu tohoto obrázku. Podobným způsobem je vygenerována nová přehledová mapa s vyznačením aktuálního výřezu. Webový server poté zkompletuje výsledný HTML kód, který pošle klientu.

## 4.5. Základní atributové informace nemovitostí

Jeden typ atributových dat již byl zmíněn - parcelní a domovní čísla, která je možno zobrazit přímo v mapě. Jestliže chce uživatel získat další informace o nemovitostech, stačí zvolit nástroj **Dotaz**, zvolit vrstvu pro dotazování a na mapě označit prvek zájmu. Nejdůležitějšími parametry odeslanými na webový server jsou nyní obrazové souřadnice bodu kliknutí. Poté, co jsou převedeny na souřadnice v S-JTSK a z parametru *qlayer* je zjištěno, do které vrstvy se ptát, je s pomocí mapového serveru realizován samotný dotaz. Jednoduše řečeno, mapový server zjišťuje, zda se na zadaných souřadnicích v určené vrstvě vyskytuje nějaký geoprvek. V případě že ano, již nic nestojí v cestě přístupu k atributům vybrané nemovitosti.

Všechny atributové informace o nemovitostech se ovšem nenachází v jediné tabulce, ale jsou rozptýleny po rozvětvené struktuře ISKN. Z toho důvodu se neobejdeme bez SQL dotazů. U parcel se ptáme na tyto informace:

- katastrální území,
- parcelní číslo,
- výměru,
- druh pozemku,
- způsob využití pozemku,
- domovní číslo případné budovy, která na parcele stojí,
- list vlastnictví, na kterém se parcela nachází.

Parcelní číslo, výměra, druh a způsob využití pozemku jsou uloženy přímo v tabulce *PAR*, na zbytek tedy použijeme SQL. Praktické provedení je zřejmé z následující ukázky kódu, který je v případě úspěšného dotazu na parcelu připojen (funkce *include*) k hlavnímu skriptu:

**Příklad 4.3. Způsob získání atributových informací o parcele**

```
// vytvoreni pripojeni k databazi
$dbconn=pg_connect("host=localhost dbname=nvf user=root password=830101")
    or die("Pojížení k databázi se nezdařilo: ".pg_last_error());

// *** získání údaje o parcele ***
// SQL dotaz
$query='SELECT ku.nazev, tel.cislo_tel, bud.cislo_bud_komplet
        FROM par JOIN katuze ku ON (ku.kod=par.katuze_kod)
            JOIN tel ON (tel.id=par.tel_id)
            LEFT JOIN bud ON (bud.id=par.bud_id)
        WHERE par.id='.$shape_id;
$result=pg_query($query) or die('Dotaz selhal: '.pg_last_error());
$row=pg_fetch_row($result); // získání výsledku
$katuze_nazev=ucwords($row[0]);
$lv_cislo=$row[1];
$cislo_domovni=$row[2];
pg_free_result($result); // uvolnění výsledku
pg_close($dbconn);      // uzavření připojení

// výpis atributu
echo "<hr>";
echo "<div>";
echo "Katastrální území: ".$katuze_nazev."<br>";
echo "Parcelní číslo: ".$my_shape->values["par_cislo_komplet"]."<br>";
echo "Výměra [m2]: ".$my_shape->values["vymera_parcely"]."<br>";
echo "Druh pozemku: ".$my_shape->values["drupoz_nazev"]."<br>";
if($my_shape->values["zpvypo_nazev"]!="") {
    echo "Způsob využití pozemku: ".$my_shape->values["zpvypo_nazev"]."<br>";
}
if(isset($cislo_domovni)) echo "Budova: ".$cislo_domovni."<br>";
echo "LV: <a href='lv.php?cislo_tel=$lv_cislo' target='_blank'>$lv_cislo</a>";
echo "</div>";
```

Atributové informace o budově jsou získávány obdobným způsobem. Jedná se o tyto údaje:

- část obce,
- domovní číslo,
- způsob využití budovy,
- číslo parcely, na které budova stojí,

- list vlastnictví, na kterém se budova nachází.

SQL dotaz byl sestaven takto:

#### Příklad 4.4. Dotaz pro získání atributových informací o budově

```
$query="SELECT c.nazev, bud.cislo_bud_komplet, zvb.zkratka,
        par.par_cislo_komplet, tel.cislo_tel
FROM bud JOIN zpvbybu zvb ON (zvb.kod=bud.zpvbybu_kod)
        JOIN par ON (par.bud_id=bud.id)
        LEFT JOIN tel ON (tel.id=bud.tel_id)
        LEFT JOIN casobc c ON (c.kod=bud.caobce_kod)
WHERE bud.id=".$my_shape->values["id"];
```

Atributové informace nemovitostí jsou zobrazeny vlevo od mapy, pod tlačítka pro vybírání vrstev. V případě neúspěšného dotazu (na místě zadaném souřadnicemi nebyl nalezen žádný geoprvek) je tato skutečnost oznámena.

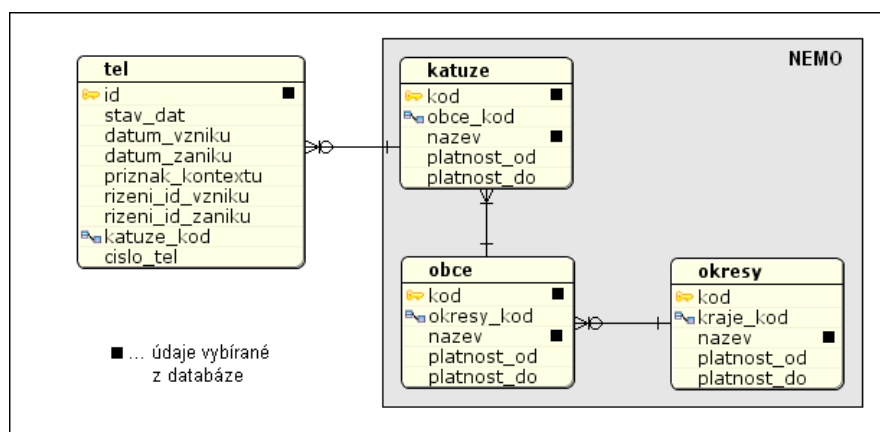
## 4.6. Sestavení listu vlastnictví

*List vlastnictví* (správně „výpis z katastru nemovitostí“) je jedním ze základních výstupů ze souboru popisných informací katastru nemovitostí. Ukládá se v katastrálním území vždy jeden pro skupinu nemovitostí nebo jednotek, ke kterým jsou evidovány shodné údaje o vlastníku a oprávněném z dalších práv [Vyhl26/07].

Pro vizualizaci listu vlastnictví, na kterém se dotazovaná nemovitost nachází, slouží odkaz v poslední řádce výpisu základních atributových informací o nemovitosti (číslo LV). Tento odkaz vede na stránku `lv.php`, ke které je v URL adrese ještě připojen parametr `cislo_tel`. Výsledná URL potom vypadá například: `lv.php?cislo_tel=872`. Skript sestavující list vlastnictví lze logicky rozdělit, stejně jako jsou rozděleny informace na listu vlastnictví do několika částí.

### 4.6.1. Záhlaví

V záhlaví jsou uvedeny údaje o datu, hodině a minutě, ke kterým výpis z katastru nemovitostí prokazuje stav evidovaný v katastru, údaje o okresu, obci a katastrálním území, číslo listu vlastnictví a informace o druhu číslování parcel. Vazby mezi tabulkami ilustruje obrázek :



**Obrázek 4.3. Schéma dotazu pro získání údajů v záhlaví**

Na základě čísla LV předaného v URL je vybráno ID listu vlastnictví (pro další použití) a ostatní údaje ze souvisejících tabulek - viz příklad:

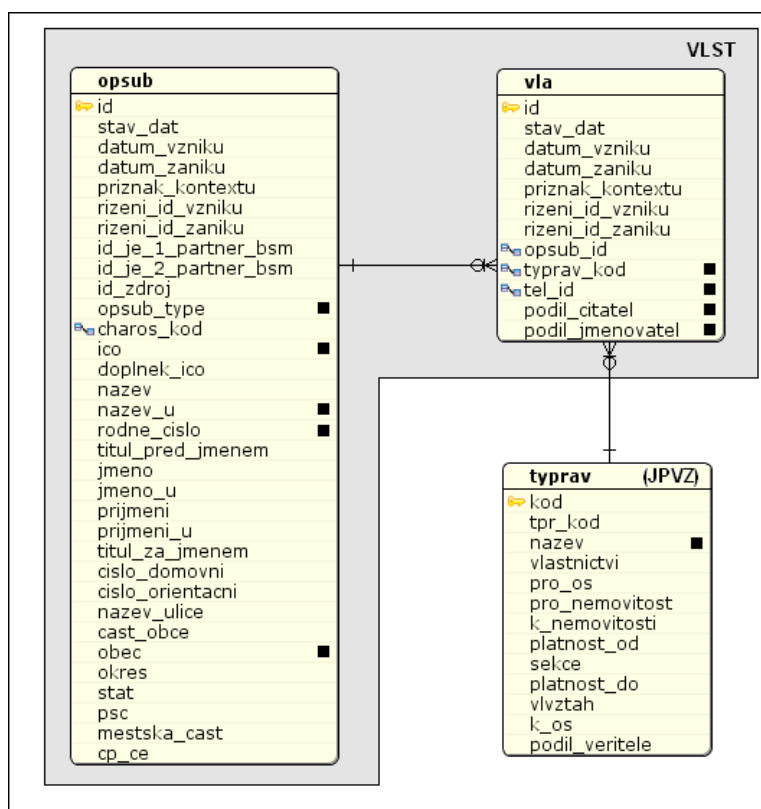
#### **Příklad 4.5. SQL dotaz na údaje v záhlaví LV**

```
$query="SELECT t.id,ku.kod,ku.nazev,ob.kod,ob.nazev,ok.nazev
FROM tel t, katuze ku, obce ob, okresy ok
WHERE t.cislo_tel=$cislo_tel AND t.katuze_kod=ku.kod
AND ku.obce_kod=ob.kod AND ob.okresy_kod=ok.kod";
```

### **4.6.2. Část A**

V části A listu vlastnictví se nachází údaje o oprávněných subjektech a vztazích, které mají k nemovitostem uvedeným v části B. Dále je případně uveden podíl na spoluvlastnictví nebo jiném právu.





Obrázek 4.4. Vztahy mezi tabulkami v části A

Vlastnické vztahy jsou vybrány na základě toho, že každý vlastnický vztah (řádek tabulky *VLA*) je vztažen k jednomu listu vlastnictví (vazba přes *TEL\_ID*), takže stačí vybrat ty řádky, které mají hodnotu *TEL\_ID* odpovídající sestavovanému listu vlastnictví:

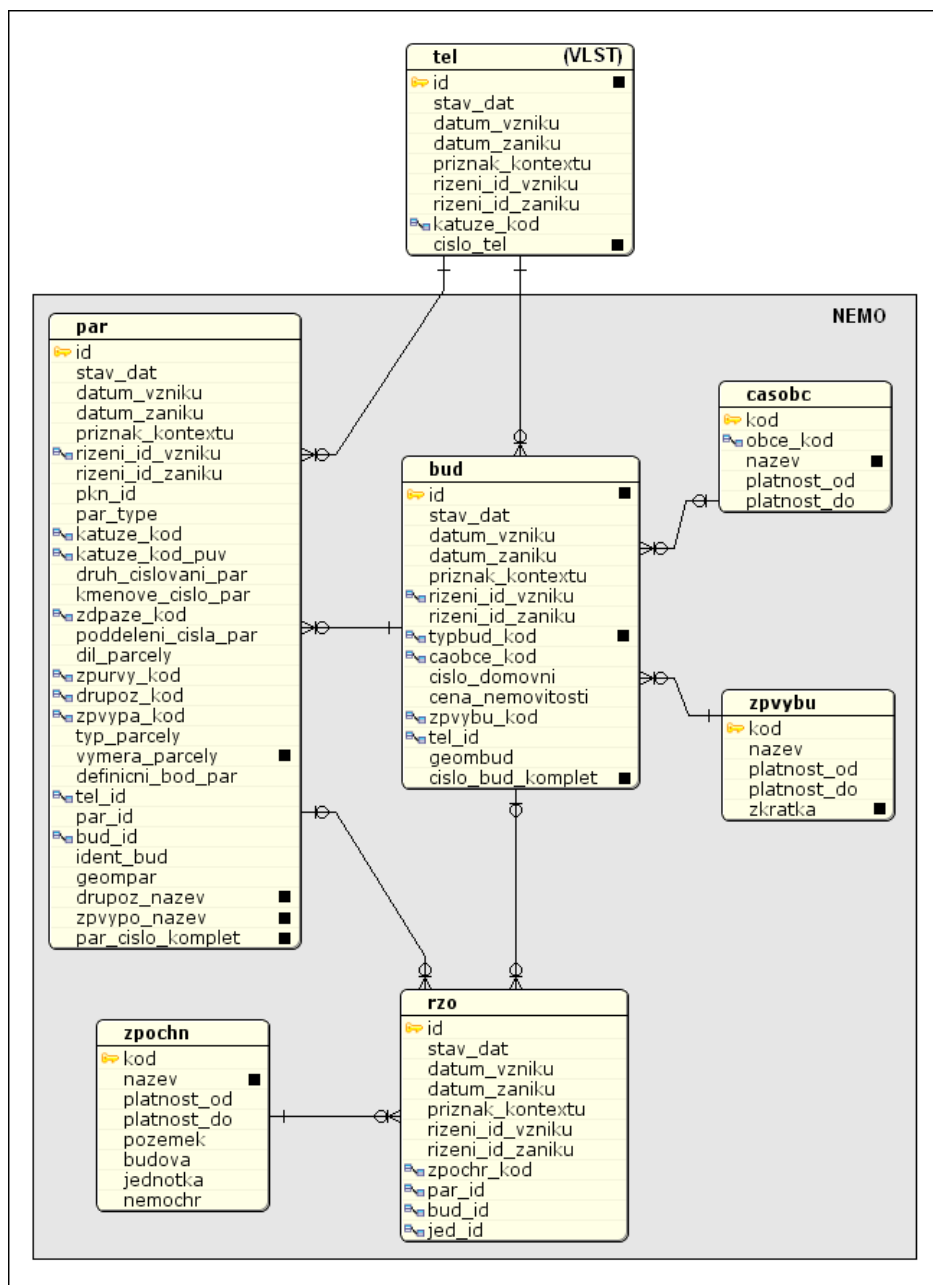
#### Příklad 4.6. SQL dotaz na údaje v části A

```
$query="SELECT v.podil_citatel, v.podil_jmenovatel, t.nazev, o.nazev_u,
        o.opsub_type, o.ico, o.rodne_cislo, o.obec, v.tel_id, v.typrav_kod
FROM vla v, typrav t, opsub o
WHERE t.kod = v.typrav_kod AND o.id = v.opsub_id AND v.tel_id=$tel_id
ORDER BY v.typrav_kod";
```

Vlastnickým vztahem není jen samotné vlastnictví, ale také např. právo hospodařit s majetkem státu, správa nemovitostí ve vlastnictví kraje apod. Proto jsou vybrané záznamy seřazeny podle kódu typu právního vztahu tak, aby stejné typy vlastnického vztahu byly pohromadě a daly se nadepsat názvem vlastnického vztahu.

### 4.6.3. Část B

Část B listu vlastnictví obsahuje údaje o nemovitostech a jednotkách, které vlastník nebo spolu-vlastníci zapsaní v části A vlastní, popřípadě ke kterým má oprávněný další právo.



Obrázek 4.5. Vztahy mezi tabulkami v části B

Výběr nemovitostí je realizován dvěma SQL dotazy - jeden vybírá parcely a druhý budovy. Díky tomu, že druh pozemku a způsob využití pozemku je nakopírován přímo v tabulce PAR (viz 4.2 – „Instalace PostGIS a import dat VF ISKN“), je dotaz na parcely o něco jednodušší.

Oba však pracují na stejném principu, vybírají parcelu, resp. budovu na základě shody sloupce *TEL\_ID* s identifikátorem sestavovaného listu vlastnictví, a k tomu připojují další potřebné tabulky.

#### Příklad 4.7. SQL dotaz pro část B - parcely

```
$query="SELECT p.par_cislo_komplet,
        p.vymera_parcely,p.drupoz_nazev,p.zpvypo_nazev,p.id,z.nazev
FROM par p LEFT JOIN rzo r ON (p.id=r.par_id)
        LEFT JOIN zpochn z ON (r.zpochr_kod=z.kod)
WHERE tel_id=$tel_id
ORDER BY druh_cislovani_par,kmenove_cislo_par,poddeleni_cisla_par";
```

#### Příklad 4.8. SQL dotaz pro část B - budovy

```
$query="SELECT c.nazev,b.tybud_kod,b.cislo_bud_komplet,zpv.zkratka,b.id,
        zpo.nazev,p.par_cislo_komplet,t.id,t.cislo_tel
FROM bud b LEFT JOIN rzo r ON (b.id=r.bud_id)
        LEFT JOIN zpochn zpo ON (r.zpochr_kod=zpo.kod)
        LEFT JOIN casobc c ON (b.caobce_kod=c.kod), zpvzbu zpv, par p, tel t
WHERE b.tel_id=$tel_id AND b.zpvzbu_kod=zpv.kod AND b.id=p.bud_id AND p.tel_id=t.id
ORDER BY tybud_kod";
```

U budov je při výpisu nutné navíc ošetřit teoretickou možnost, že budova stojí na více pozemcích. V tom případě by se ve výsledcích dotazu objevila vícekrát, pokaždé na jiné parcele, ale správně má být zapsána jen v jedné řádce, s příslušnými parcelami oddělenými čárkami.

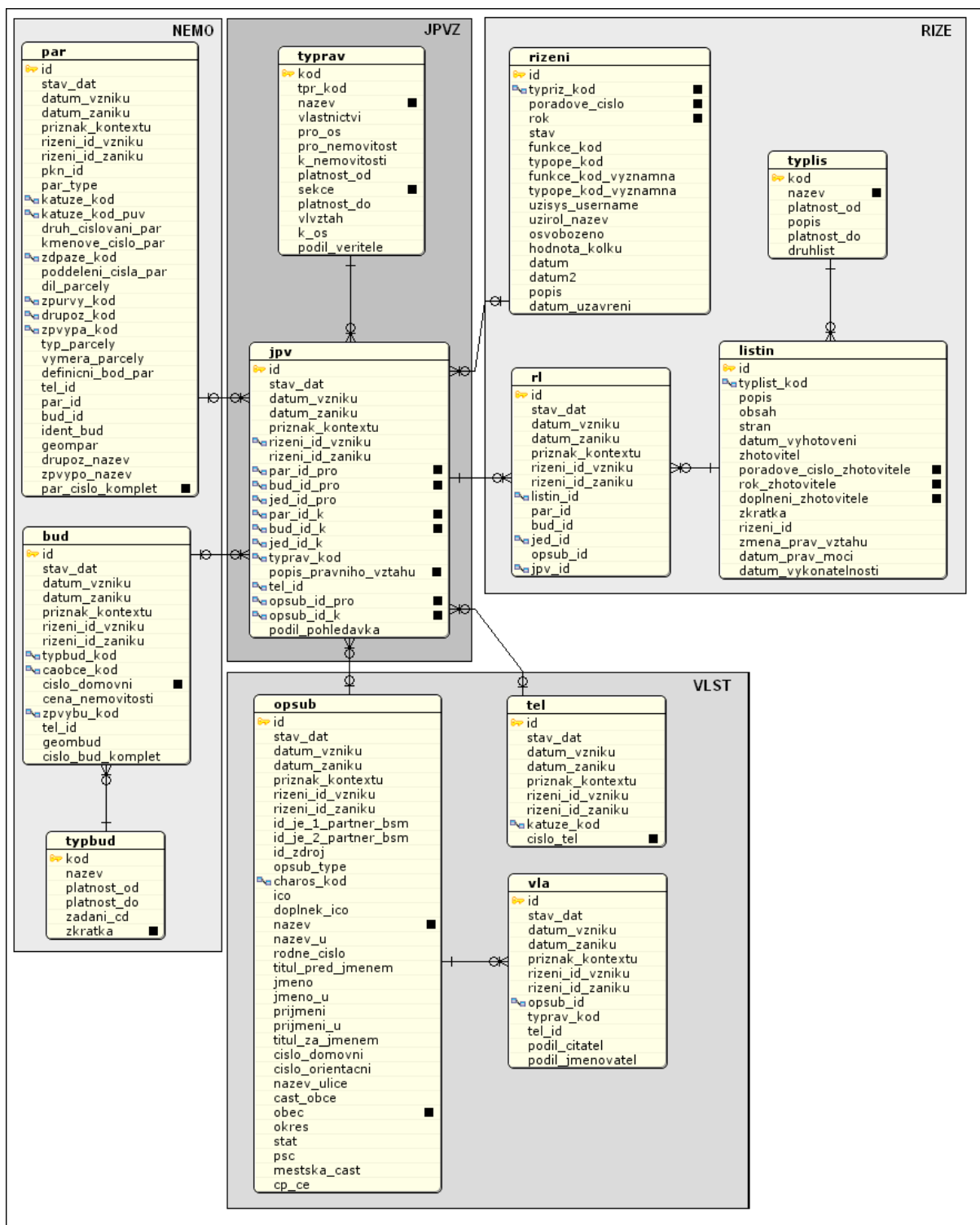
#### 4.6.4. Části B1, C, D

V části B1 se nachází věcná práva odpovídající věcnému břemeni a práva zapsaná podle dřívějších předpisů ve prospěch vlastníka nemovitostí a jednotek uvedených v části B, včetně údajů o těchto právech dotčených nemovitostech, označení protokolu a pořadové číslo, pod kterým byl zápis v dané věci proveden, a údaje o listinách, které byly podkladem pro zápis.

Část C obsahuje omezení a poznámky o skutečnostech, které vlastníka, oprávněného z dalších práv nebo třetí osobu omezují v nakládání s nemovitostmi a jednotkami uvedenými v části B, zatížení nemovitosti a jednotky jiným věcným právem, včetně údajů o dotčené nemovitosti a jednotce, údajů o oprávněném z jiného věcného práva, váže-li se omezení nebo jiné věcné právo k takové osobě, označení protokolu a pořadové číslo, pod kterým byl zápis v dané věci proveden, a údaje o listinách, které byly podkladem pro zápis.

V části D najdeme poznámky o podaném žalobním návrhu, informace o zahájení pozemkových úprav a další údaje, které se váží k osobě uvedené v části A nebo k nemovitosti uvedené v části B, včetně údajů o tímto zápisem dotčené nemovitosti nebo jednotce, údajů o oprávněném z dalších práv uvedeném v části A, váže-li se tato poznámka nebo další údaj katastru k takové osobě, označení protokolu a pořadové číslo, pod kterým byl zápis v dané věci proveden, a údaje o listinách, které byly podkladem pro zápis [Vyhl26/07].

Informace pro části B1, C a D jsou získávány až na drobnosti stejným postupem, ve všech případech se jedná o jiný právní vztah, jenž má záznam v tabulce *JPV*. Z toho důvodu jsou data pro tyto části získávána v jednom dotazu a o příslušnosti záznamu do sekce je rozhodováno až ve skriptu, na základě údaje ve sloupci *SEKCE* tabulky *TYPRAV*.



Obrázek 4.6. Vztahy mezi tabulkami v části B1, C, D

Vzhledem k tomu, že jiné právní vztahy se mohou týkat mnoha věcí (oprávněného subjektu, nemovitosti, jiného právního vztahu - viz 3.2.3 – „Jiné právní vztahy (JPVZ)“), musí dojít ke

spojení poměrně značného množství tabulek. Z této struktury jsou potom vybrány ty záznamy, u nichž platí některá z následujících podmínek:

- právní vztah je přímo svázán s listem vlastnictví (v *JPV* je vyplněno *TEL\_ID* a odpovídá),
- právní vztah se týká nemovitosti nacházející se na tomto listu vlastnictví (odpovídající *TEL\_ID* v tabulce *PAR* či *BUD*),
- právní vztah se týká oprávněného subjektu, který má vlastnický vztah k tomuto listu vlastnictví (odpovídající *TEL\_ID* v tabulce *VLA*).

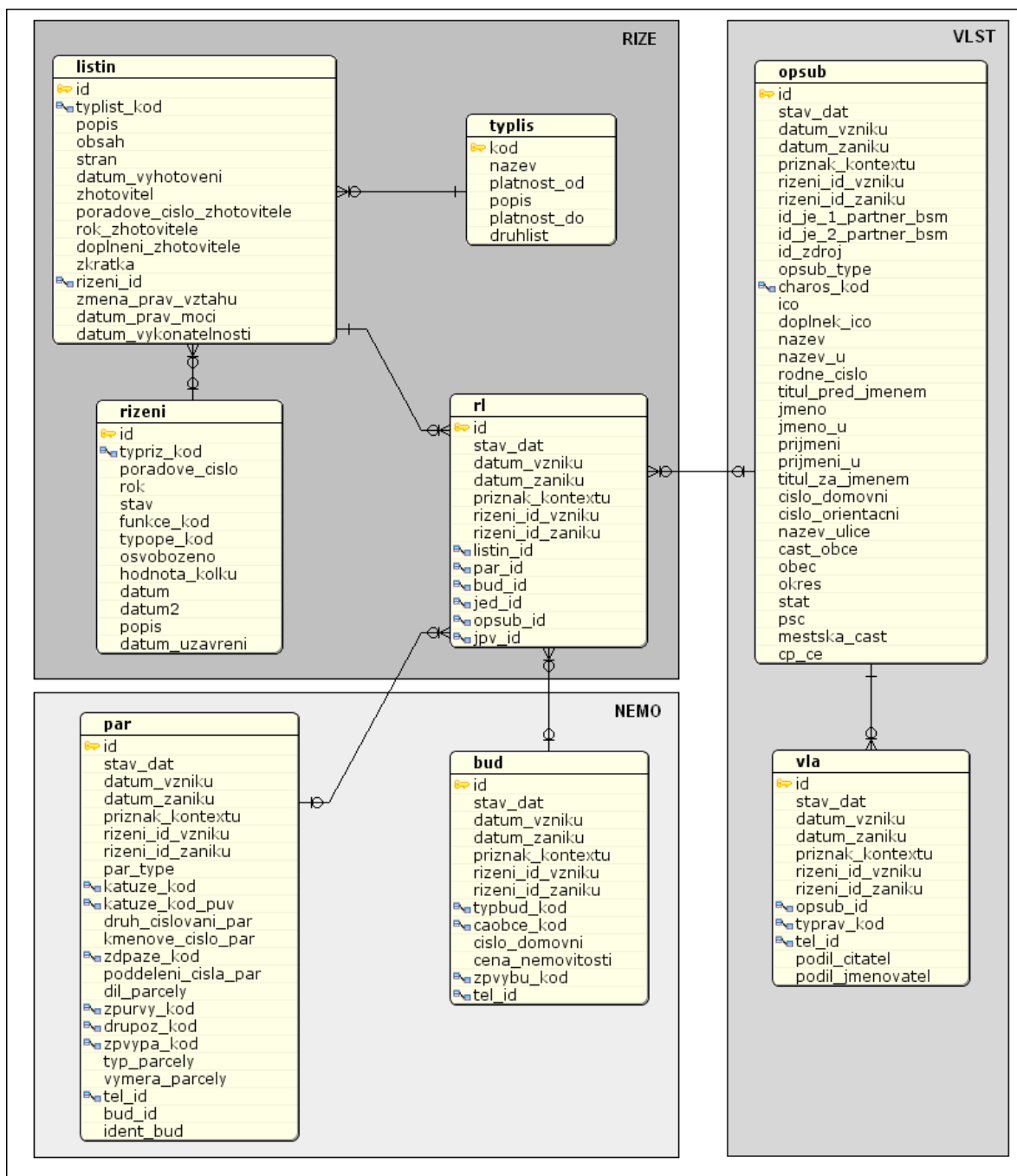
#### Příklad 4.9. SQL dotaz pro části B1, C, D

```
$query="SELECT t.nazev,jpv.popis_pravniho_vztahu, r.typriz_kod,r.poradove_cislo,
        tel.cislo_tel, jpv.opsub_id_k, jpv.bud_id_k, jpv.par_id_k,
        jpv.opsub_id_pro, jpv.bud_id_pro, jpv.par_id_pro,
        opsub.nazev, tb.zkratka, bud.cislo_domovni,
        par.par_cislo_komplet, opsub.obec, typlis.nazev,
        t.sekce, r.rok, listin.poradove_cislo_zhotovitele,
        listin.doplneni_zhotovitele, listin.rok_zhotovitele
FROM jpv LEFT JOIN par ON (jpv.par_id_k=par.id OR jpv.par_id_pro=par.id)
LEFT JOIN bud ON (jpv.bud_id_k=bud.id)
LEFT JOIN typbud tb ON (tb.kod=bud.typbud_kod)
LEFT JOIN opsub ON (opsub.id=jpv.opsub_id_k OR opsub.id=jpv.opsub_id_pro)
LEFT JOIN vla ON (vla.opsub_id=opsub.id)
LEFT JOIN tel ON (tel.id=jpv.tel_id)
JOIN typrav t ON (jpv.typrav_kod=t.kod)
JOIN rizeni r ON (jpv.rizeni_id_vzniku=r.id)
LEFT JOIN rl ON (rl.jpv_id=jpv.id)
LEFT JOIN listin ON (listin.id=rl.listin_id)
LEFT JOIN typlis ON (typlis.kod=listin.typlist_kod)
WHERE jpv.tel_id=$tel_id
      OR (jpv.tel_id IS NULL AND(par.tel_id=$tel_id
      OR bud.tel_id=$tel_id OR vla.tel_id=$tel_id))
ORDER BY t.kod";
```

Výsledek dotazu se prochází řádek po řádku a záznamy jsou zařazeny do příslušné sekce. Podle toho jsou poté ze záznamu vybrány atributy, které se budou zobrazovat.

## 4.6.5. Část E

Tato část obsahuje údaje o listinách, které byly podkladem k zápisu vzniku nebo změny vlastnického práva a dalších práv uvedených v části A.



Obrázek 4.7. Vztahy mezi tabulkami v části E

Způsob výběru záznamů je podobný jako v jiných právních vztazích - jsou vybrány listiny vázané na nemovitosti nebo oprávněný subjekt nacházející se na listu vlastnictví. Vazbu mezi listinami a nemovitostmi, případně listinami a oprávněnými subjekty zprostředkovává vazební tabulka *RL*.

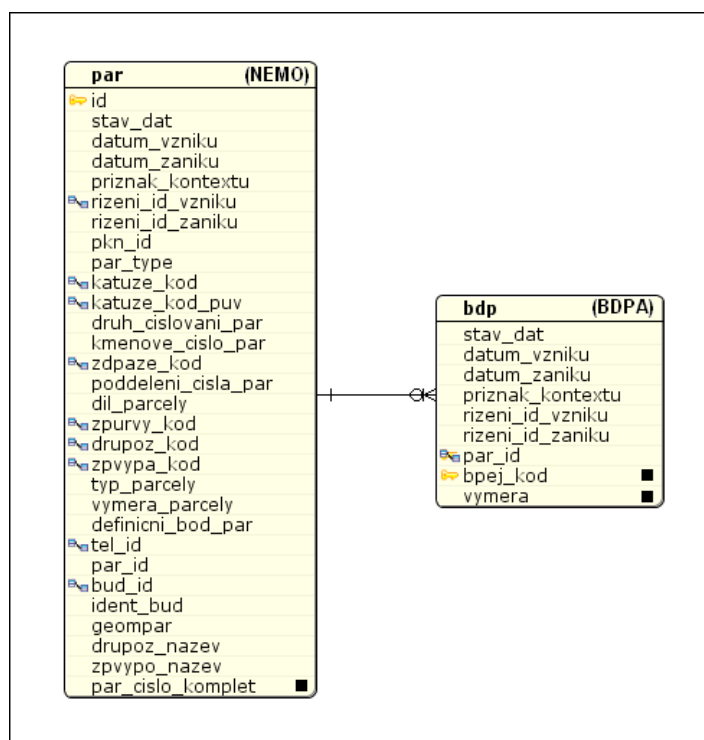
#### Příklad 4.10. SQL dotaz pro část E

```
$query="SELECT DISTINCT list.popis, typlis.nazev,
        os.nazev, os.obec, os.rodne_cislo, os.ico,
        riz.typriz_kod, riz.poradove_cislo,
        riz.rok, list.poradove_cislo_zhotovitele,
        list.doplneni_zhotovitele, list.rok_zhotovitele
FROM listin list JOIN typlis ON (typlis.kod=list.typlist_kod)
JOIN rl ON (rl.listin_id=list.id)
LEFT JOIN par ON (par.id=rl.par_id)
LEFT JOIN bud ON (bud.id=rl.bud_id)
LEFT JOIN jpv ON (jpv.id=rl.jpv_id)
LEFT JOIN opsub os ON (os.id=rl.opsub_id)
LEFT JOIN vla ON (vla.opsub_id=os.id)
JOIN rizeni riz ON (riz.id=list.rizeni_id)
WHERE par.tel_id=$tel_id OR bud.tel_id=$tel_id
      OR vla.tel_id=$tel_id
ORDER BY riz.rok";
```

#### 4.6.6. Část F

V části F se nachází údaje o vztahu BPEJ k parcelám zemědělských pozemků uvedeným v části B. Tabulka *BDP* je vazební tabulkou realizující vazbu M:N mezi tabulkami *PAR* a *BPEJ*. Tabulka *BPEJ* však není pro výpis atributových informací části F třeba, protože všechny nutné údaje se nachází v tabulce *BDP*. Prvek této tabulky si lze představit jako díl parcely, který má definován kód BPEJ a výměru.





Obrázek 4.8. Vztahy mezi tabulkami v části F

## Příklad 4.11. SQL dotaz pro část F

```

$query="SELECT bdp.vymera, bdp.bpej_kod, par.par_cislo_komplet
FROM par, bdp
WHERE par.tel_id=$tel_id AND bdp.par_id=par.id
ORDER BY par.druh_cislovani_par, par.kmenove_cislo_par, par.poddeleni_cisla_par";
  
```

Všechny informace získané z databáze jsou nakonec zobrazeny uživateli na samostatné HTML stránce, generované PHP. Pro formátování výstupu jsou použity kaskádové styly (CSS, viz 2.2.1 – „Klient“).

# Kapitola 5

## Závěr

Cílem této práce bylo implementovat mapový server pro vizualizaci dat katastru nemovitostí. Vstupní data - vzorové soubory výměnného formátu ISKN - byla naimportována do připravené databáze PostGIS pomocí skriptu vytvořeného J. Orálkem. Do tohoto skriptu byly později doprogramovány dvě nové funkce, které sestavují parcelní, resp. domovní čísla rovnou při importu a ukládají je do nově vytvořeného sloupce - díky tomu se může parcelní či domovní číslo zobrazovat přímo na mapě.

Pro vizualizaci dat byl vybrán produkt UMN MapServer s rozhraním PHP MapScript, které umožňuje mnohem větší interaktivitu aplikace než CGI program. PHP, jako skriptovací jazyk na straně serveru, generuje HTML výstup, takže stránka získává větší dynamiku. Seznámení se s tímto mapovým serverem proběhlo nad očekávání snadno, zásluhu na tom má hlavně dobrá dokumentace, tutoriály a ochotní lidé v diskuzních skupinách.

Zásadní částí práce, jak se časem ukázalo, bylo získávání atributových dat a hlavně sestavování listu vlastnictví. Struktura výměnného formátu ISKN je sice popsána, což dává představu i o vztazích mezi tabulkami ISKN, horší situace je ale ohledně listu vlastnictví, jehož podrobná struktura popsána není. U některých prvků je na první pohled zřejmé, co představují, často ale bylo nutné hledat odpovídající tabulku a sloupec v databázi metodou pokus - omyl. Pro posouzení správnosti výstupů sloužila aplikace *Dálkový přístup do KN na zkoušku*, která obsahuje stejná vzorová data.

Fáze sestavování listu vlastnictví byla tedy nakonec časově nejnáročnější, díky čemuž se již nepodařilo realizovat několik dalších vylepšení. Nabízí se například možnost vyhledávání parcel a budov na základě čísla, zvýraznění nemovitostí jednoho listu vlastnictví v mapě nebo odkazy z výpisu listu vlastnictví zpět do mapy či jiné listy vlastnictví. Zřejmě by bylo vhodné optimalizovat SQL dotazy a zkusit lépe využít možností databáze PostGIS. V neposlední řadě by se dalo přemýšlet o lepším ovládní aplikace (použití některého z hotových produktů, například Map) a návrhu celkového designu aplikace.

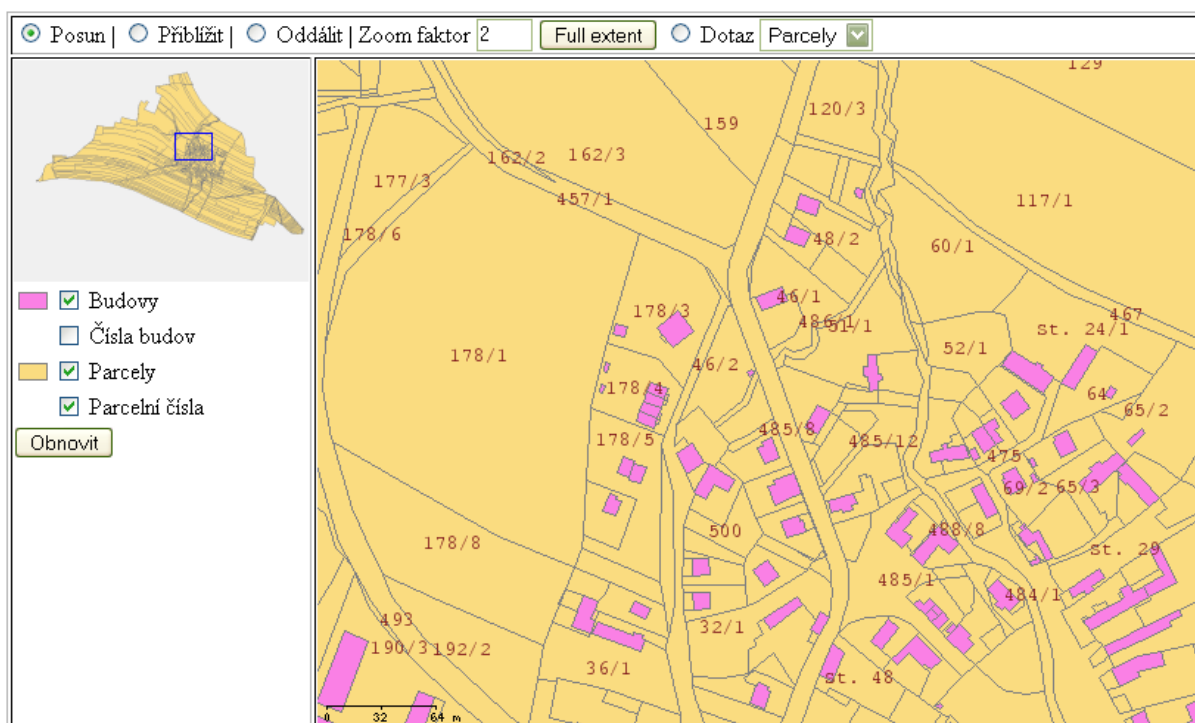
# Literatura

- [i-Apa] *The Apache HTTP Server Project [online]*. Copyright © 1999-2005 The Apache Software Foundation. <<http://httpd.apache.org>>.
- [i-ARC] *ARCDATA PRAHA - co je GIS [online]*. Copyright © 1992-2006 ARCDATA PRAHA, s. r. o.. <<http://www.arcdata.cz/uvod/co-je-gis>>.
- [i-Gar2005] Jesse James GARRET. *Ajax: A New Approach to Web Applications [online]*. 18. 2. 2005. <<http://www.adaptivepath.com/publications/essays/archives/000385.php>>.
- [i-Kos2006] Jiří KOSEK. *XML a nové trendy v publikování na Webu [online]*. Copyright © 2000-2006 Jiří Kosek. <<http://www.kosek.cz/vyuka/izi228/prednasky/xml/foil10.html>>.
- [i-MS-Ex] Vinko VRSALOVIC. *PHP/Mapscript By Example [online]*. 12. 12. 2005. <<http://mapserver.gis.umn.edu/docs/howto/phpmapscript-byexample/>>.
- [i-OSS] Společnost pro výzkum a podporu Open Source. *Co je svobodný software? [online]*. 10. 11. 2005. <<http://www.oss.cz/co-je-to-svobodny-software>>.
- [i-Skl2006] Radek SKLENIČKA. *Interoperabilita v GIS podle specifikací OGC [online]*. <[http://geoinformatics.fsv.cvut.cz/wiki/index.php/Gi2006\\_-\\_Interoperabilita\\_v\\_GIS\\_podle\\_specifikac%C3%AD\\_OGC](http://geoinformatics.fsv.cvut.cz/wiki/index.php/Gi2006_-_Interoperabilita_v_GIS_podle_specifikac%C3%AD_OGC)>.
- [i-Sni2005] Martin SNÍŽEK. *AJAX - kde jsou hranice? [online]*. 13. 9. 2005. <<http://www.snizekweb.cz/clanky/ajax-kde-jsou-hranice/>>.
- [i-TS] *Terminologický slovník zeměměřictví a katastru nemovitostí [online]*. Copyright © 2005-2006 VÚGTK. <<http://www.vugtk.cz/termkom/indtk.html>>.
- [ČÚZK2007] Český úřad zeměměřický a katastrální. *Struktura výměnného formátu informačního systému katastru nemovitostí České republiky*. 27. 3. 2007.
- [ESRI] ESRI. *ArcIMS 9 Architecture and Functionality*. An ESRI White Paper. 2004. Copyright © 2004 ESRI.
- [Jed2006] Jan JEDLINSKÝ. *Způsoby uložení prostorových dat v databázi pro účely pozemkového datového modelu*. Diplomová práce. Západočeská univerzita v Plzni. 2006.

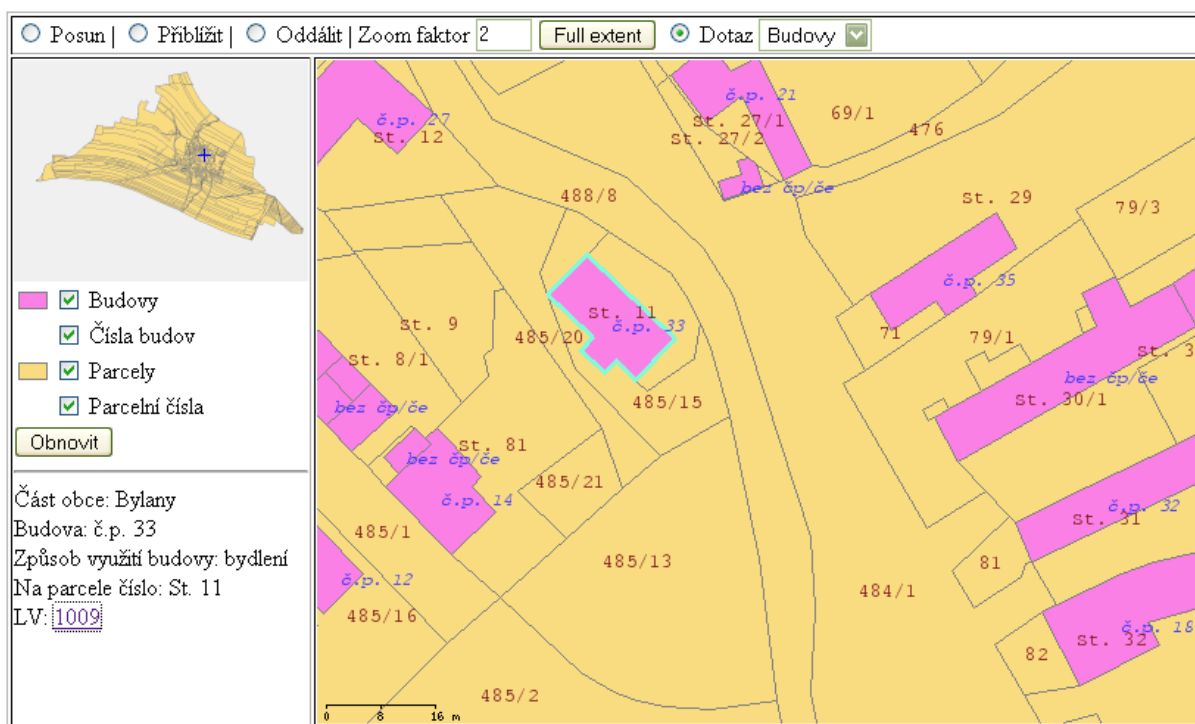
- 
- [Kol2004] Milan KOLLINGER. *Návrh a implementace finančně nenáročného způsobu publikace geografických dat v síti Internet*. Diplomová práce. Západočeská univerzita v Plzni. 2004.
- [Ora2006] Jakub ORÁLEK. *Možnosti využití nekomerčního geografického software pro tvorbu prostorového rozhraní informačního systému malé obce*. Diplomová práce. Západočeská univerzita v Plzni. 2006.
- [Pen2003] Zhong-Ren PENG a Ming-Hsiang TSOU. *Internet GIS*. Distributed Geographic Information Services for the Internet and Wireless Networks. John Wiley & Sons, Inc.. Hoboken, New Jersey. 2003. ISBN 0-471-35923-8.
- [Tuč1998] Ján TUČEK. *Geografické informační systémy*. Principy a praxe. Computer Press. Praha. 1998. ISBN 80-7226-091-X.
- [TSG] Úrad geodézie, kartografie a katastra Slovenskej republiky a Český úřad zeměměřický a katastrální. *Terminologický slovník geodézie, kartografie a katastra*. ÚGKK SR. Bratislava. 1998. ISBN 80-88716-36-5.
- [Vyhl26/07] *Vyhláška 26/2007 Sb - Katastrální vyhláška, §17*.

# Příloha 1

## Grafické ukázky z aplikace



Obrázek 1.1. Uživatelské rozhraní aplikace



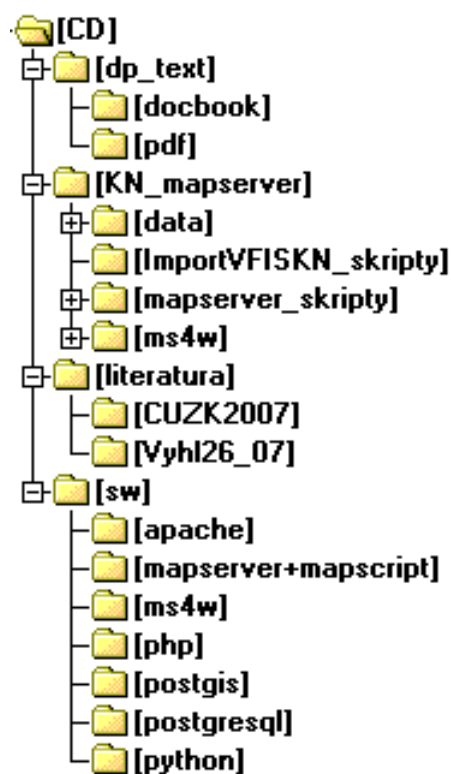
**Obrázek 1.2. Výpis atributových informací budovy**

VÝPIS Z KATASTRU NEMOVITOSTÍ			
prokazující stav evidovaný k datu 25.05.2007 04:34:47			
Okres: Chrudim	Obec: 571245 Bylany		
Kat. území: 616567 Bylany	List vlastnictví: 1009		
A	Vlastník, jiný oprávněný vlastnictví	Identifikátor	Podíl
	<b>kopr julius, bylany</b>	<b>520308481</b>	
B	Nemovitosti		
	Pozemky		
	Parcela	Výměra [m2]	Druh pozemku
			Způsob využití
			Způsob ochrany
	st. 11	307	zastavěná plocha a nádvoří
	73/1	9473	ostatní plochy
	414/13	2475	orná půda
	503	560	zahrada
			manipulační plocha
			zemědělský půdní fond
			zemědělský půdní fond
	Budovy		
	Typ budovy		
	Část obce, č. budovy	Způsob využití	Způsob ochrany
	Na parcele		
	Bylany, č.p. 33	bydlení	St. 11
B1	Jiná práva	- Bez zápisu	
C	Omezení vlastnického práva	- Bez zápisu	
D	Jiné zápisy		
	Typ vztahu		
	Vztah pro	Vztah k	
1	změna výměr obnovou operátu		
		LV: 1009	Z-1700498/1999
1	změna číslování parcel		
	parcela utvořena z p.p.73 pk		
		Parcela: 73/1	Z-1700498/1999
E	Nabývací tituly a jiné podklady zápisu		
	Listina		
1	darovací smlouva v12 5960/1997 z 27.5.1997, právní účinky vkladu vznikly dnem 28.11.1997		
		POLVZ: 96/1998	Z-1700096/1998
	Pro: kopr julius, bylany		RČ/IČO: 520308481
F	Vztah bonitovaných půdně ekologických jednotek (BPEJ) k parcelám		
	Parcela	BPEJ	Výměra [m2]
	414/13	30200	391
	414/13	35900	2084
	503	31000	560
	Pokud je výměra bonitních dílů parcel menší než výměra parcely, zbytek parcely není bonitován		
	Jiří Petrák, ZČU v Plzni	Vyhотовeno: 25.05.2007 04:34:47	
	<b>Autor neručí za správnost a úplnost údajů!</b>	Vyhотовil: Vyhотовeno KN_mapserverem	
	Řízení PÚ: .....	Podpis, razítko:	

Obrázek 1.3. Sestavený list vlastnictví

## Příloha 2

### Obsah přiloženého CD



dp_text	docbook	- zdrojový text DP v Docbooku
	pdf	- text DP v tisknutelné podobě
KN_ma- pserver	data	- data VF ISKN verze 2.8 a 3.0
	ImportVFISKN_skrity	- skripty v Pythonu pro import dat VFK do PostGIS
	mapserver_skrity	- zdrojové soubory mapového serveru
	ms4w	- zdrojové soubory mapového serveru připravené pro použití v MS4W



literatura		- výběr z použité literatury
sw	apache	- instalace Apache HTTP Server 2.2.3 pro Win32
	mapserver+mapscript	- distribuce UMN MapServer 4.8.1 a PHP MapScript pro PHP verze 5.1.2
	ms4w	- distribuce MS4W (MapServer for Windows) verze 2.2.3
	php	- distribuce PHP verze 5.1.2
	postgis	- instalace PostGIS 1.2.1 pro PostgreSQL 8.2
	postgresql	- instalace PostgreSQL 8.2.4
	python	- instalace jazyka Python 2.5 a rozšíření PyWin32