VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INTELIGENT SYSTEMS

OBJEKTOVĚ ORIENTOVANÉ GEOGRAFICKÉ DATABÁZE

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE Bc. JURAJ KOMLOŠI
AUTHOR

BRNO 2010

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INTELIGENT SYSTEMS

OBJEKTOVĚ ORIENTOVANÉ GEOGRAFICKÉ DATABÁZE

OBJECT ORIENTED GEOGRAPHIC DATABASES

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE — Bc. JURAJ KOMLOŠI
AUTHOR

VEDOUCÍ PRÁCE — Ing. MARTIN HRUBÝ, PhD.
SUPERVISOR

BRNO 2010

Abstrakt

Tato práce demonstruje použití objektově orientovaných geografických databází. Vychází z geografických informačních systémů. Popisuje jejich základní vlastnosti a možnosti použití v praxi. V práci jsou popsány základní vlastnosti a metody objektově orientovaného použití. Ty jsou demonstrovány na geografických informačních systémech. Je zaveden pojem objektově orientovaný rastr a objektově orientovaný vektor a způsob uložení jednotlivých prvků v objektově orientované databázi. Návrh způsobu ukládání dat do geografické objektově-orientované databáze je podložen případovou studií.

Abstract

This paper demonstrates object oriented geographic databases's applying. It is based on geographic information systems. It describes properties of geographical information systems and their uses in practise. There are also described main properties and methods of object oriented standards in this paper. All of them are used in geographical oriented systems. New ideas are defined, object oriented raster, object oriented vector and some methods defining storing these items in object oriented database. Proposal for a method of storing data in a geographical object-oriented database is supported by case study.

Klíčová slova

Geografický informační systém, GIS, vektor, rastr, objektově orientovaná databáze, VisualWorks, Gemstone.

Keywords

Geographic information systém, GIS, vector, raster, object oriented database, VisualWorks, Gemstone.

Citace

Komloši Juraj: Objektově orientované geografické databáze, diplomová práce, Brno, FIT VUT v Brně, 2010.

Objektově orientované geografické databáze

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně pod vedením Ing. Martina Hrubého, PhD. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

……
Juraj Komloši
19.5.2010

Poděkování

Moje poděkování patří zejména vedoucímu práce Ing. Martinu Hrubému PhD., za vedení a odborné znalosti.

Obsah

Zoznam obrázkov

Úvod

Geografické informačné systémy patria medzi najdynamickejšie sa rozvíjajúce informačné systémy. Tento pojem zoskupuje databázové, analytické, vizualizačné a štatistické metódy pri riešení zložitých interpretácií modelu reality. Geografické informačné systémy sú využívané v mnohých odvetviach, či už sú to inžinierske siete, doprava, územné plánovanie alebo armáda. Každé z týchto odvetví kladie na geografické informačné systémy špecifické požiadavky, ktoré je nutné interpretovať tak, aby zachytili realitu dôležitú z pohľadu používateľa.

Z čoraz väčším rozvojom geografických informačných systémov je potrebné riešiť otázku ukladania geografických dát do databáz. Zachytenie reality s veľkým množstvom detailov predstavuje enormné množstvo dát, ktoré je nutné ukladať a organizovať. Ukladanie dát zabezpečujú systémy správy databázy (DBMZ), ktorých úlohou je ukladať, organizovať a spravovať údaje. V súčasnosti existuje viacero rôznych DBMS architektúr. Zatiaľ sa ukazujú ako najužitočnejšie relačné DBMS. V relačnej databáze sa údaje ukladajú do tabuliek. Stĺpce spoločné pre rôzne tabuľky sa využívajú na ich vzájomné prepojenie. Táto prekvapujúco jednoduchá koncepcia je široko rozšírená, najmä pre svoju pružnosť a uplatnenie v množstve aplikácií.

Na druhej strane sú to objektovo orientované DBMS. Ich výhodou je najmä potenciál priblížiť realitu modelovaného sveta bližšie k používateľovi, aby ju dokázal jednoduchšie pochopiť. Objektovo orientovaný DBMS predstavuje využitie všetkých výhod, ktoré ponúka aj objektovo orientované programovanie. Ide najmä o prirodzenejšiu reprezentáciu zložitých dátových štruktúr, dedičnosť a polymorfizmus.

Ta ako samotné geografické informačné systémy, tak aj ich objektovo orientovaná reprezentácia je v štádiu vývoja. Cieľom tejto práce je ozrejmiť samotný pojem geografický informačný systém, ukázať opodstatnenosť použitia objektovo orientovaných geografických informačných systémov, analyzovanie metód objektovo orientovaného modelovania priestorových objektov a navrhnúť objektovo orientovaný spôsob pre ukladanie dát do databáz.

Hlavný dôraz sa bude klásť na spôsob uloženia dát v objektovo orientovanej databáze. Prínosom bude navrhnutie objektovo orientovaného modelu a implementácia prototypu postaveného na danom modeli. Ako vhodné implementačné prostredie poslúži objektovo orientovaný jazyk Smalltalk, resp. vývojové prostredie VisualWorks spolu s objektovo orientovanou databázou Gemstone. Na základe navrhnutého objektovo-orientovaného GIS systém je urobená prípadová štúdia pre jeho použitie.

1 Úvod do GIS systémov

V teoretickej časti diplomovej práce sú podrobne popísané geografické informačné systémy, objektovo-orientované GIS systémy a objektovo-orientované databázy. Každá kapitola je popísaná do podrobnejších detailov, aby si bolo možné utvoriť obraz o rozoberanej problematike.

1.1 Geografické informačné systémy

V súčasnosti existuje niekoľko definícii geografického informačného systému (GIS). Všeobecne je možné definovať GIS vyplývajúc zo samotného názvu. Geo – znamená, že GIS pracuje s informáciami a dátami, ktoré sa vzťahujú buď priamo k Zemi alebo k jej povrchu, prípadne k podpovrchovým častiam, u ktorých je známa ich lokalizácia. Grafický – znamená, že GIS spracované dáta prezentuje grafickou formou. Poskytuje grafické rozhranie pre prácu s daným výstupom. Informačný – znamená, že ide o zber, ukladanie a spracovanie údajov, z ktorých je možné získať nové informácie potrebné pre modelovanie reálneho sveta. Systém – znamená, že ide o integráciu hardvéru spolu so softvérom a priestorovými údajmi.

Definícia GIS podľa [1] hovorí, že geografický informačný systém je možné definovať ako súbor prostriedkov pre zber, ukladanie, vyhľadávanie, transformáciu, analyzovanie a zobrazovanie priestorových údajov z reálneho sveta z hľadiska:

- ich polohy vzhľadom k definovanému súradnicovému systému,
- ich popisných - atribútových vlastností,
- ich priestorových vzťahov k iným objektom, ich topológie.

GIS podobne ako každý informačný systém poskytuje databázu ako úložisko veľkého množstva dát, analýzy týchto dát, databázové operácie a štatistiky dát. Navyše umožňujú prezentáciu uložených a analyzovaných dát cez mapové rozhranie prostredníctvom mapy. Dáta, ktoré spracováva GIS, sa nazývajú priestorové dáta. Sú to dáta, ktoré sa vzťahujú k určitým miestam v priestore a ktoré majú známu lokalizáciu, teda majú určenú geografickú polohu. Za priestorové dáta môžeme považovať mestá, rieky, pohoria, to znamená, že všetkým vieme priradiť polohu na mape. Využitie GIS nájdeme v mnohých oblastiach. V obchode, inžinierskych sieťach, štátnej správe, doprave, telekomunikáciách, územnom plánovaní ale aj archeológií a v mnohých ďalších.

1.1.1 Zber údajov

Jednou z najdôležitejších častí GIS-u je zber údajov. Zber údajov je ešte aj v súčasnej dobe pomerne drahou záležitosťou. Dôvodom sú vysoké nároky na presnosť, správnosť a úplnosť zberaných údajov. Pred zberom údajov je nutné zamyslieť sa nad typom údajov, ktoré chceme zozbierať a následne analyzovať. Tak isto musíme brať do úvahy typ GIS aplikácie, pre ktorú sú dané údaje určené. Zber údajov môžeme rozdeliť podľa typu údajov na:

- geometrické údaje, ktoré obsahujú aj topologické vzťahy,
- popisné údaje, ktoré popisujú skúmané geografické objekty.

Geometrické údaje môžeme chápať ako grafickú bázu. Teda súhrn všetkých grafických informácií uložených v geografickom informačnom systéme. Grafická báza je tvorená predovšetkým digitálnou mapovou bázou a ostatnými grafickými informáciami doplňujúcimi mapovú časť. Z pohľadu informačného systému hovoríme o grafických informáciách, buď vektorového alebo rastrového typu. Popisné údaje naopak reprezentujú negrafickú databázu. Ide o databázu, ktorá obsahuje popisné textové a číselné informácie o skúmaných objektoch, ktoré sú určené pre daný GIS. Z informačného hľadiska ide o dáta, ktoré sa používajú na označenie atribútov objektov.

Ďalšie kritérium rozdelenie dát je ich pôvod. Dáta rozdeľujeme na základe ich pôvodu na:

- *primárne* – predstavuje zber údajov priamo v teréne, ide predovšetkým o geodetické metódy, fotogrametické metódy a laserovú altimetriu,
- *sekundárne* – predstavuje nepriamy zber údajov, kedy vychádzame z už zozbieraných údajov, medzi tieto metódy môžeme zaradiť skenovanie, digitalizáciu.

Výsledkom zberu dát sú dáta vo forme vektorov alebo vo forme rastrov.

1.1.2 Vektory

Pri vektorovej reprezentácii údajov sú všetky objekty uchovávané vo forme bodov, línií, reťazec línií, plôch a povrchov. Spomenuté geometrické prvky slúžia na reprezentáciu zložitejších typov údajov. Pre správne a najmä logické uloženie údajov do databázy si podrobne opíšeme.

Bod (ang. point) je v priestore reprezentovaný pomocou priestorových súradníc, jeho dimenzia v priestore je 0. Medzi ďalšie informácie obsiahnuté v bode môžeme zaradiť informácie o napojení bodu v línií. Bod môže byť samostatný, t.j. nenapojený v línií, medziľahlý bod, t.j. jeden z bodov tvoriacich líniu, alebo koncový.

Línia (ang. line) je tvorená postupnosťou susediacich úsečiek, ktoré sú spojené v medziľahlých bodoch. Táto postupnosť začína v počiatočnom bode a končí v koncovom bode. Dimenzia línií v priestore je jedna.

Reťazec línií (ang. polyline) je geometrický prvok, v ktorom je každá línia obsiahnutá práve jedenkrát. Všetky uzly sa vyskytujú presne v dvoch líniách, okrem prvého a posledného uzla v reťazci. V prípade uzatvoreného reťazca línií sa aj prvý a posledný uzol vyskytuje práve v dvoch líniách. Dimenzia reťazca línií v priestore je 1.

Plochu (ang. area) je definovaná ako uzavretá líniá, prípadne reťazec línií, kde sú identické prvý aj posledný uzol. Dimenzia plochy v priestore je 2.

Povrch (ang. surface) je plocha, ktorá má priradené body, v každom jej bode, to znamená aj vo vnútorných bodoch. Podobne ako plocha má aj povrch dimenziu v priestore 2.

Definovanie samotných objektov nám v geografických informačných systémoch nestačí. Dôležité sú vzťahy medzi nimi. Kým človek si pri pohľade na dané objekty dokáže odvodiť ich vzájomné vzťahy, počítaču ich musíme definovať. Vzájomné vzťahy medzi objektmi sú predmetom topológie. Topológia predstavuje matematický spôsob, akým je možné vyjadriť priestorové vzťahy medzi objektmi. Základné vzťahy, ktoré sú predmetom topológie, sú:

- susednosť – každá línia je orientované, má smer a obsahuje informácie o objektoch susediacich naľavo a napravo,
- konektivita – dve línie sa spájajú v uzloch,
- orientácia – línia má smer, je orientovaná,
- definícia plochy – množina línií uzatvárajúce nejakú plochu definujú polygon.

Takmer všetky vektorové dátové štruktúry sú založené na bodoch. Body majú definovanú polohu pomocou súradníc. Z týchto bodov vychádzajú línie a vytvárajú zložitejšie objekty ako sú napríklad polygóny. Existuje viacero topologických modelov – špagetový, základný topologický a hierarchický. Podrobne si v ďalšej podkapitole popíšeme základný topologický model.

1.1.2.1 Základný topologický model

Základný topologický model predstavuje najčastejšie používaný model slúžiaci na zachytávanie priestorových vzťahov medzi objektmi. Pravidlá sú nasledovné. Každá línia začína aj končí v uzle (bode). Línie sa môžu pretínať, pričom ich priesečník je opäť uzol. Jednotlivé uzly tvoriace línie sú uložené v databáze s presnými hodnotami súradníc. Aby boli zachované úplné topologické vzťahy, tak línie obsahujú ešte identifikátor ľavého, resp. pravého polygónu.

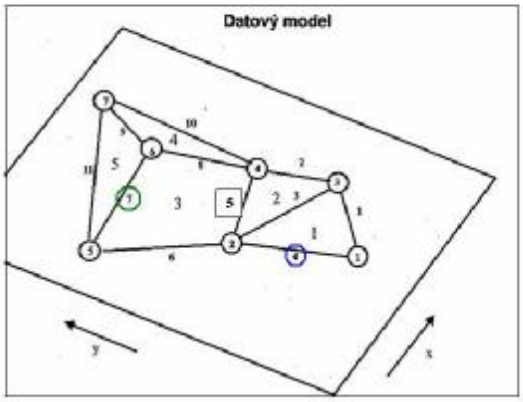
Medzi nevýhody tohto modelu patrí nepochybne fakt, že pri vyhľadávaní konkrétnej línie je nutné prejsť celú databázu línií. Ak ide o polygon tak je nutné prejsť dané záznamy niekoľkokrát. Príklad topológie je uvedený na obr. 1.1.

Soubor topologických vztahů

Hrana	Pravý Polygon	Levý Polygon	Počátek v bodě	Konec v bodě
1	1	0	3	1
2	2	0	4	3
3	2	1	3	2
4	1	0	1	2
5	3	2	4	2
6	3	0	2	5
7	3	5	5	6
8	3	4	6	4
9	4	5	7	6
10	0	4	7	4
11	5	0	5	7

Soubor souřadnic bodů

Uzel	X souřadnice	Y souřadnice
1	23	8
2	17	17
3	29	15
4	26	21
5	8	26
6	22	30
7	24	36



Obr. 1.1: Základná topológia vektorov [2].

1.1.3 Rastre

Reprezentácia rastrov sa predovšetkým zameriava na určité územie ako celok. Používa sa na zobrazenie spojito sa meniacich javov, ako je napríklad teplota ovzdušia alebo typ pôdy. Základom každého rastra je najmenšia jednotka nazývajúca sa bunka (ang. cell). Študovaná plocha sa rozdelí na sieť spravidla pravidelných buniek. Ich zoskupenie tvorí mozaiku. Každá rastrová bunka zaznamenáva informáciu alebo atribút , ktorý vyjadruje stav v danej lokalite. Bunky nemusia byť pravidelné, môžu nadobúdať tieto tvary:

- štvorcový,
- trojuholníkový,
- hexagonálny.

Dôvody prečo sa najčastejšie používa pravidelná štvorcová mriežka sú prosté. Tento typ mriežky je kompatibilný s kartézskym súradnicovým systémom, je kompatibilný s mnohými zariadeniami určenými pre vstup a výstup dát.

Výhody trojuholníkovej mriežky spočívajú v zachytení buniek s rôznou orientáciou. Využíva sa pri modelovaní digitálneho modelu terénu, kedy je možné zachytiť aj jednotlivé sklony skúmaného terénu. Umožňuje to vlastnosť každej bunky, ktorá si okrem súradníc zachováva aj veličinu výška, t.j. každému bodu je priradená jeho výška. To umožňuje zistiť veľkosť sklonu ako aj jeho smer. Nevýhodou trojuholníkovej mriežky sú zložitejšie algoritmy, ktoré analyzujú daný model terénu. V poslednej, hexagonálnej mriežke, stojí za pozornosť vlastnosť, že stredy všetkých buniek sú od seba rovnako vzdialené, tzv. ekvidistantné.

Okrem spomínaného delenia sa rastre delia na:

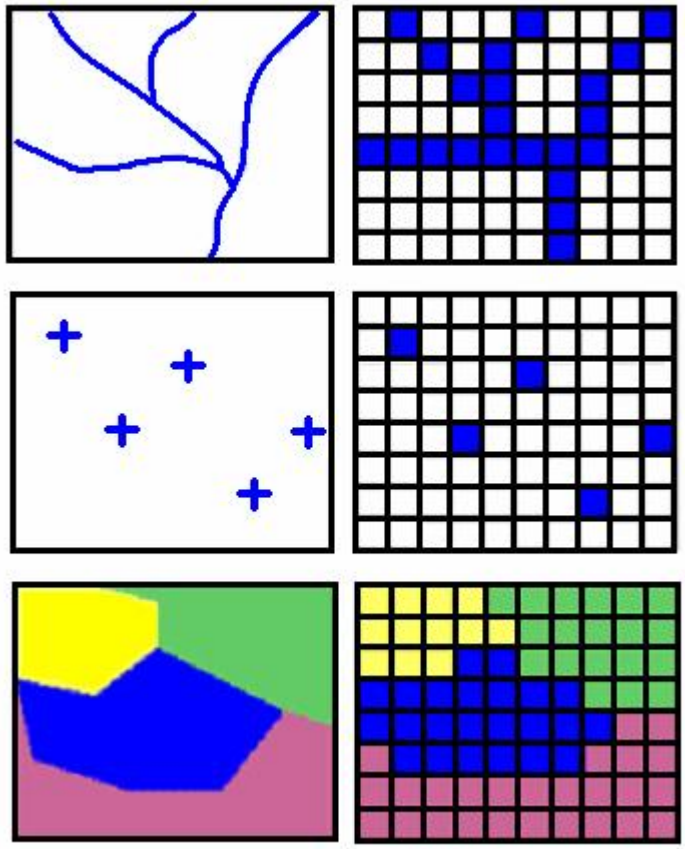
1. *pravidelné* – všetky bunky majú rovnakú veľkosť aj tvar, ich výhodou je oveľa jednoduchšie spracovávanie a ukladanie údajov, nevýhodou je nie celkom presné zachytenie danej lokality (spravidla terén nepozostáva z pravidelných jednotiek),
2. *nepravidelné* – bunky majú rôznu veľkosť aj tvar, ich výhodou je oveľa reálnejšie zachytenie reality (možnosť nížiny s pohoriami, resp. údoliami), no na druhej strane ich nevýhodou je implementácia zložitých algoritmov.

Topológia rastrov vyplýva z ich štruktúry. Keďže ide o mriežku obsahujúcu bunky je pomerne zrejmé, ktoré bunky susedia s danou bunkou. Ich pozícia v stĺpci, resp. riadku sa dá určiť podobne ako je to v maticiach. Podrobnejšie si rozoberieme pravidelnú štvorcovú mriežku v nasledujúcej podkapitole.

1.1.3.1 Pravidelná štvorcová mriežka

Pravidelná štvorcová mriežka patrí medzi veľmi často sa využívajúce dátové štruktúry. Jednotlivé bunky mriežky zachytávajú informáciu dvoma spôsobmi. Každá bunka môže obsahovať hodnotu buď vo svojom strede alebo v celej ploche bunky. Tak ako pri vektorovej reprezentácií, tak aj v rastrovej je možné zachytiť body, línie a plochy, viď obr. 1.2. Bod je reprezentovaný jednou bunkou, línia je reprezentovaná radou spojených buniek s rovnakou hodnotou a plocha je reprezentovaná skupinou susediacich buniek s rovnakou hodnotou. V realite sa pomerne často vyskytuje prípad kedy bunka mriežky nenesie žiadnu hodnotu. V tom prípade sa danej bunke priradí predvolená hodnota, ktorá má svoj význam v celom systéme.

Pri zobrazovaní rastrovej mriežky je dôležité zvoliť vhodné rozlíšenie, ináč povedané vzťah rastrová bunka – pixel. V prípade, že sa zvolí príliš veľké rozlíšenie existuje riziko straty určitých hodnôt. Naopak, ak sa zvolí príliš malé rozlíšenie, tak pochopiteľne uloženie takejto štruktúry zaberie oveľa viac miesta na disku.



Obr. 1.2: Reprezentácia bodu, línie a plochy v rastroch [3].

1.2 Objektovo orientovaný GIS

Predošlé kapitoly obsahovali definíciu GIS so všetkými jeho charakteristickými črtami. Teraz sa zameriame na jeho objektovo orientovanú reprezentáciu. Aké sú výhody objektovo orientovaného GIS v porovnaní s relačným? Všeobecné vzaté ide najmä o sofistikovanejšie zachytenie modelovanej reality a snahu priblížiť daný model čo najviac používateľovi. Vychádza sa pri tom zo základných prvkov objektovo orientovaného prístupu.

1.2.1 Objektovo orientovaný prístup

Objektovo orientovaný prístup je založený na práci s *objektom* (ang. object) ako základným elementom, z ktorého sa skladá skúmaný systém a navrhovaný projekt. Objektom môže byť aj abstraktná jednotka alebo štruktúra, ktorá nám napomôže modelovať skúmanú časť univerza. Objekt

je súčasťou programu, objektovo orientované programy sa skladajú len z objektov, čím je zabezpečený unifikovaný prístup, správanie, možnosť dedenia, agregácie a podobne [4]. Objekt vlastní okrem svojej identifikácie (OID - Object ID), svojho mena aj svoje atribúty (členské dáta, attributes) a metódy (členské funkcie, methods). Vyznačuje sa istým správaním (behaviour) voči ostatným objektom a okoliu. Abstrakciou objektov s rovnakými vlastnosťami je trieda. Správanie a služby objektu zabezpečujú metódy, stav objektu zabezpečujú jeho hodnoty uchovávané v členských dátach – atribútoch. Medzi základné vzťahy objektov patria:

- príslušnosť každého objektu k určitej triede – reprezentovaná vzťahom IS-A, objekty sú inštanciami tried,
- dedičnosť – reprezentovaná vzťahom IS-KIND-OF, triedy môžu medzi sebou dediť určité spoločné črty a modifikovať ich,
- zloženosť objektu z častí – reprezentovaný vzťahom IS-PART-OF, objekt sa môže skladať z iných objektov.

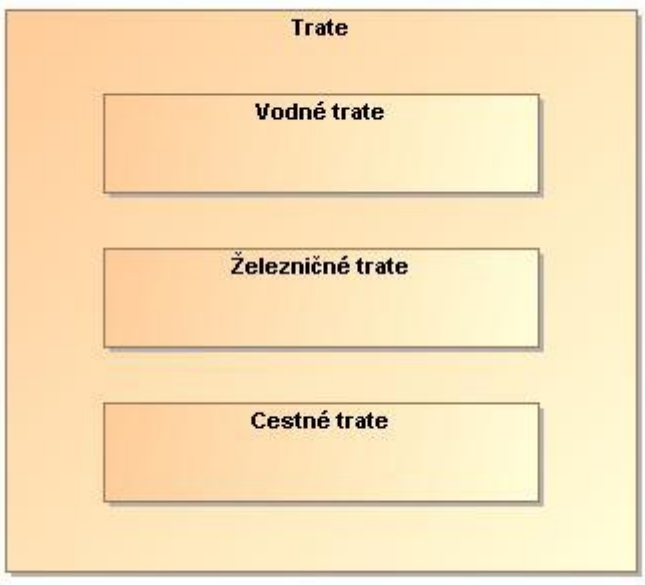
Klasifikácia

Klasifikácia je priradenie objektov do tried. Objektom sa chápe inštancia triedy, preto sa zvykne používať aj názov tohto vzťahu INSTANCE-OF. Objekt má svoje atribúty a špecifikované správanie k svojmu okoliu. Každá inštancia má je popísaná operáciami, pomocou ktorých je možné manipulovať s objektmi. Všetky objekty patriace tej istej triede majú rovnaké vlastnosti a operácie, ktoré nimi môžu manipulovať. Ako príklad môžeme uviesť model krajiny, ktorá pozostáva z tried parcely, rieky, jazerá. Inštanciou triedy rieky môže byť rieka Hron. Každý typ objektu ma pridelené vlastnosti (atribúty) a metódy. Pre inštanciu triedy rieky to môže byť počet vodných elektrárni na danej rieke. Inštancia jazerá môže obsahovať atribút aktuálny objem vody znázornené na obr. 1.3. Z uvedených vlastností vyplýva, že jednotlivé objekty patriace do tej istej triedy sa líšia hodnotami svojich atribútov, teda ich vlastnosťami. Každá rieka má iný počet vodných elektrárni, jazerá majú rôzny objem vody a podobne.

Obr. 1.3: Príklad klasifikácie geoobjektov.

Generalizácia

Generalizácia spája niekoľko tried objektov so spoločnými metódami do špecifickejších tried, takzvaných supertried [5]. Všetky objekty v danej supertriede sú vo vzťahu is-a, niekedy sa využíva označenie vzťahu rodič – potomok, kde rodič je supertrieda a potomok je jeho podtrieda. Generalizácia sa používa na znázornenie závislosti podtriedy od supetriedy. Neznamená to, že supertrieda a podtrieda popisujú dva rôzne objekty, je to stále ten istý objekt. Ako príklad uvediem supertriedu trate, ktorá obsahuje podtriedy vodné trate, železničné trate a cestné trate. Cestná trať s označením I. triedy je zároveň inštanciou triedy trate aj cestné trate. Pri generalizácii supertriedy môže obsahovať jednu alebo viacero podtried. Ďalej je vhodné pouvažovať nad obmedzeným množstvom úrovni pre ktoré je daná trieda nadtriedou pre konkrétnejšie triedy. Príkladom je nadtrieda trate obsahujúca podtriedu vodné trate, ktorá ďalšie podtriedy umelé vodné trate a prírodné vodné trate. Kým vodné trate sú supertriedou pre umelé a prírodné vodné trate, trieda trate je supertriedou pre vodné trate a zároveň aj pre triedy umelé a prírodné trate znázornené na obr. 1.4.



Obr. 1.4: Príklad generalizácie geoobjektov.

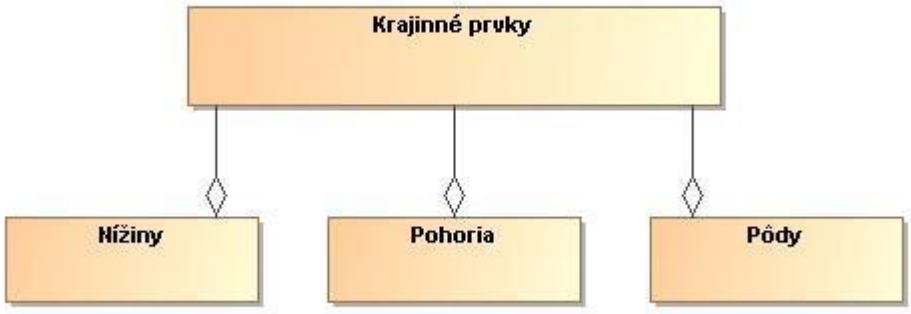
Asociácia

V asociácií sú spojené dva alebo viacero nezávislých objektov a vzťah medzi objektmi sa považuje za objekt vyššej úrovne [6]. Spájané objekty sa nazývajú členmi. Príkladom asociácie v GIS je napríklad susednosť, kedy určitá parcela je spojená s riekami, ktoré cez ňu pretekajú. V asociácii sa

vyzdvihujú vlastnosti danej väzby pričom vlastnosti spájaných objektov sú menej významné. Inštancia väzby môže byť rozdelená medzi inštancie objektov, ktoré spája. Spravidla sa pre všetky objekty vo väzbe aplikujú operácie pre každého jej člena jednoduchou slučkou – loop.

Agregácia

Je to mechanizmus abstrakcie, podobný asociácii, v ktorom model pozostáva z objektov, ináč povedané objekty pozostávajú z ďalších objektov [7]. V modeli môže byť niekoľko objektov spojených do objektu vyššej úrovne. Sú to agregované alebo kompozitné objekty, v ktorých každá časť má svoju vlastnú funkcionalitu. Operácie nad agregovanými objektmi nie sú kompatibilné s operáciami ich častí. Platí to aj opačne. Podobne ako v asociácii sú v agregovanom objekte menej významné vlastností objektov, z ktorých je zložený. A takisto platí, že agregovaný objekt môže byť rozdelený na objekty, z ktorých pozostáva. Operácie nad agregovaný objektom môžu byť aplikované na jeho časti buď sekvenčne alebo paralelne. Ako príklad zobrazený na obr. 1.5 je použitá agregovaná trieda krajinné prvky pozostávajúca z tried pôdy, pohoria, nížiny. Tieto triedy sú časťou triedy krajinné prvky a trieda krajinné prvky pozostáva z nich.



Obr. 1.5: Príklad agregácie krajinných prvkov.

Dedičnosť

Dedičnosť je metóda, ktorá definuje triedu z hľadiská jednej alebo viacerých iných všeobecnejších tried [8]. To znamená podtrieda môže zdediť niektoré vlastnosti a metódy od svojej jednej alebo viacero nadtried. Vlastnosti, ktoré sú spoločné pre triedu aj jej podtriedu sa definujú iba raz, v nadtriede. Podtrieda môže mať naviac definované ďalšie vlastnosti aj metódy. Tie však nie sú dostupné pre objekty nadtriedy na rozdiel od vlastností nadtriedy, ktoré sú dostupné aj pre podtriedy. Z pohľadu algebry je nadtrieda prienikom všetkých axióm z jej podtried. Dedičnosť má vlastnosť

tranzitivity. Všetky operácie nadtriedy môžu používať jej podtriedy, podtriedy danej podtriedy atď. Rozlišujeme dva typy dedičnosti – jednoduchú a viacnásobnú.

Jednoduchá dedičnosť znamená, že každá trieda má maximálne jednu podtriedu a opačne. V praxi sa v zložitejších systémoch používa menej ako viacnásobná.

Viacnásobná dedičnosť znamená, že trieda môže mať viac podtried a tie ďalšie podtriedy. Nejedná sa o hierarchiu dedičnosti, pretože jedna trieda môže mať viac nadtried. Najjednoduchšie to je vysvetlené na obr. 1.6. Vychádzajúc z obr. 1.4., kde trate sú rozdelené na vodné, železničné a cestné, je pridané trieda vodstvo obsahujúca triedy jazerá a vodné trate. Ako je vidieť trieda vodné tratededí vlastnosti od nadtried trate a vodstvo. Táto hierarchia neplatí pre všetky objekty triedy vodstvo, pretože napríklad trieda jazerá nemá nadtriedu trate. Viacnásobná dedičnosť je veľmi užitočnou metódou v GIS, pretože výrazne zjednodušuje pohľad a samotné modelovanie reality a tak isto redukuje kód pri implementovaní. Je to kvôli definovaniu vlastností a metód v nadtriedach.

Obr. 1.6: Príklad dedičnosti v GIS systémoch.

1.2.2 Objekty v GIS

Z vlastností uvedených v predchádzajúcej kapitole je možné popísať objekty v geografických informačných systémoch a ich vlastností. Sú to:

- **druh geografického objektu** – navyše umožňuje aj špecializáciu každého druhu, napríklad cesty -> vodné cesty -> umelé vodné cesty,
- **príslušnosť objektu k druhu** - každý objekt prislúcha k určitému typu, napríklad línia je priradená k typu cesta, čím získava hneď informácie o druhu ale aj informácie o danej triede, do ktorej patrí línia,
- **unikátne vlastnosti objektu** – každý objekt má definované vlastnosti, vo svojich atribútoch, vo svojej triede, ktoré určujú okrem iného aj jeho vzťah k ostatným okolitým objektom.

1.3 Objektovo orientovaná databáza

Po definovaní základných vlastností a čŕt objektovo orientovaného prístupu v GIS, si uvedieme definíciu objektovo orientovanej databázy a požiadaviek na takúto databázu v GIS. Objektovo orientovaný databázový model predstavuje pomerne nový prístup k modelovaniu databázy. Objektovo orientovaná databáza poskytuje logický pohľad na údaje a takisto aj na informácie od nich odvodené na vyššej úrovni. Tieto informácie sú totožné s pohľadom ľudí na dané informácie. Jej úlohu je integrovať princípy skúmania do geografickej databázy [9].

Z predchádzajúcich kapitol vyplýva, že základným prvkom objektovo orientovaného modelu je objekt. Každý objekt v databáze musí obsahovať údaje o polohe, čase a téme. Tieto údaje sú reprezentované prostredníctvom atribútov a vzťahov medzi jeho priestorovými, resp. nepriestorovými zložkami. Atribútom môže byť nejaká priestorová alebo nepriestorová charakteristika. Priestorové zložky určujú polohu objektu v danom priestore pomocou atribútov. Ide napríklad o súradnice objektu, typ objektu z hľadiska tvaru – bod, čiara apod. Dôležitou priestorovou zložkou je topológia, teda vzťahy objektu k ostatným objektom. Medzi nepriestorové zložky radíme informácie o danom objekte, ktoré sa netýkajú priestoru. Zaraďujeme medzi ne napríklad individuálne fyzikálne a chemické vlastnosti objektu. Takisto ako v priestorovej zložke sú reprezentované atribútmi daného objektu. Medzi nepriestorové vzťahy objektu radíme operátory is a kind of – je druh, part of - časť niečoho, composed of – zložený z atď. Posledná zložka je časová. Vyjadrujeme ju časovými atribútmi a časovými vzťahmi. Každému objektu sa môže priradiť dátum vzniku, čas trvania, prípadne iný časový údaj. Ako časové vzťahy uvažujeme vzťahy určujúce prítomnosť objektu v danom modeli. Napríklad bol, bude, je. Všetky tieto zložky objektov, ktorých funkciou je okrem určenia polohy, času a témy objektu, zabezpečiť aj dostatok informácií na definovanie jedinečnej identity objektu vrátane vzťahov s ostatnými objektmi a charakteristiky správania, možno v zmysle [10] rozdeliť do šiestich skupín:

- originálny identifikátor,
- polohové informácie,
- nepriestorové atribúty,
- topologické vzťahy,
- netopologické vzťahy,
- metódy (metódy správania sa objektu pri činnostiach ako vytváranie nových objektov, dopytov, výpočtov, zobrazení a pod.).

1.3.1 Požiadavky na geografickú objektovo orientovanú databázu

Geografická objektovo orientovaná databáza by mala spĺňať niekoľko požiadaviek. Medzi inými perzistenciu, dotazovanie, paralelizmus, distribuovanosť a spoľahlivosť.

Perzistencia

Medzi najdôležitejšie vlastností všetkých databázových systémov patrí perzistencia. Všeobecne by sme mohli definovať perzistentnosť nasledovne. Je to vlastnosť systému trvalo udržiavať dáta, a to nielen počas jeho behu ale aj po skončení programu. Takéto dáta sa nazývajú perzistentné, zatiaľ čo dočasné dáta, dostupné len počas behu programu, sa nazývajú tranzientné. Kvalitný databázový systém by mal podporovať ako tranzientné tak aj perzistentné dáta a prácu s nimi.

V objektovo orientovaných databázových systémoch sa perzistencia týka udržovania ľubovoľných objektov rôznych tried. Možným problémom v spomínaných systémoch je výskyt zložitejších dátových štruktúr v porovnaní s relačným modelom. Jednotlivé štruktúry sú medzi sebou previazané veľkým množstvom odkazov, resp. referencií. Preto je dôležité aplikovať správny typ perzistencie. Poznáme niekoľko druhov perzistencie, z ktorých je pre objektovo orientovanú databázu najvhodnejšia ortogonálna perzistencia. Tento typ perzistencie zaisťuje ukladanie dát v rovnakom tvar ako v pamäti. Ďalej zabezpečuje:

- korektné ukladanie, otváranie a načítanie dát po malých častiach, zväčša podľa potreby,
- rovnaký systém perzistencie pre kód aj dáta.

Dotazovanie

Koncoví používatelia, ktorí pracujú s GIS systémami, nemusia mať rozsiahle znalosti o danom databázovom systéme, aby mohli vykonávať rôzne dotazovacie úlohy. Práve GIS systémy sú charakteristické množstvom dotazov nad databázou. Vyhľadávanie rôznych geoobjektov, ich vzájomných vzťahov, prípadne zložitejšie dotazy nad priestorovými dátami. Databázový systém by

mal preto podporovať jazyk na takej úrovni, aby umožnil aj laickému používateľovi vyhľadávanie dát podľa jednoduchých kritérií. Spravidla sa jedná o štandardizovaný dotazovací jazyk.

Paralelizmus

Geografická objektovo orientovaná databáza musí spĺňať požiadavku na dotazovanie. Rešpektujúc zložitosť dotazov nad priestorovými údajmi ďalšou požiadavkou objektovo orientovaného databázového systému je paralelizmus. Pre rýchlejšiu analýzu dát by bolo vhodné výpočet paralelizovať. Paralelizmus by mohol prebiehať na základe typu analyzovaných dát, prípadne ich lokality v danom modeli. Ideálnym stavom by bol stav, kedy by sa dali paralelizovať všetky operácie a analýzy nad spracovávanými dátami. Okrem spomínaných vlastností, paralelizmus nepriamo súvisí aj s distribuovateľnosťou systému. To znamená, že paralelizmus by mohol prebiehať na lokálnych, resp. vzdialených serveroch naraz.

Distribuovanosť

V GIS systémoch podobne ako aj v mnohých ďalších pristupujú viacerí používatelia z viacerých miest naraz. Takisto konkrétne dáta, s ktorými pracuje GIS systém sú distibuované na viacerých miestach a sú spojené navzájom pomocou siete. V praxi to znamená, že získané geodáta sú uložené v mieste ich zberu, prípadne v miestach odkiaľ sú najčastejšie dotazované. Napríklad miestne samosprávy najčastejšie riešia otázky týkajúce sa územia, ktoré oni spravujú a z veľkej pravdepodobnosti sa na ostatné územia dotazujú oveľa menej. Preto údaje o danom území budú uložené v blízkosti ich zozbierania. Výhodou je samozrejme rýchlosť dotazovania, odľahčenie komunikačnej siete apod. Medzi nevýhody tohto prístupu radíme zložitejší komplexnejší návrh systému, definovanie a aplikovanie sieťových protokolov, ako aj možnosť vzniku chýb pri prenose dát z jednej stanice na druhú.

Spoľahlivosť

Nemenej dôležitou požiadavkou pre databázový systém je jeho spoľahlivosť. Keďže pracuje s množstvom dát, v GIS aj pomerne drahých dát, mal by byť schopný ochrániť ich pred poškodením alebo celkovou stratou. Zlyhať môže hardvér aj softvér. Zlyhaniu zo strany hardvéru vie predísť databázový systém len čiastočne, a to zálohovaním dát do záložnej externej pamäte. Účinnejšou formou ochrany je ochrana pred zlyhaním softvéru. Ide o transakčný mechanizmus, ktorý umožňuje zachovať konzistentosť databázy. Transakcia sa môže vykonať buď úspešne celá alebo sa nevykoná vôbec. Ak nastane chyba systém musí zabezpečiť návrat do posledného konzistentného stavu.

2 Analýza modelovania GIS systémov

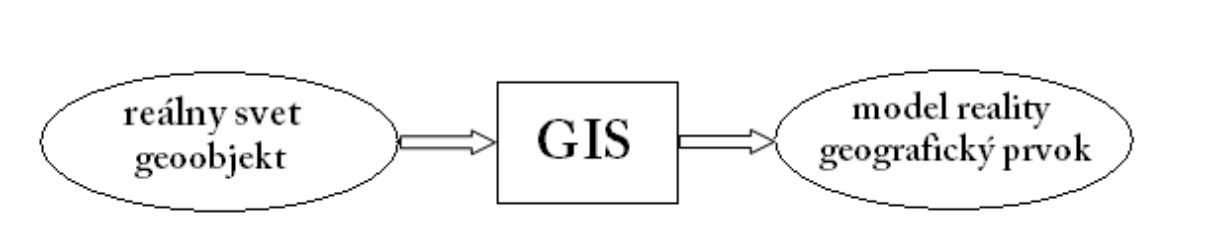
Kapitola analýzy sa zaoberá analyzovaním modelovania geografických objektov reálneho sveta. Obsahuje podkapitolu modelovanie reality z geografického hľadiska, v ktorej som sa snažil priblížiť spôsob, akým je možné modelovať geoobjekty a faktory, ktoré modelovanie najviac ovplyvňujú. Súčasťou tejto kapitoly je aj prípadová štúdia, v ktorej je zanalyzovaný súčasný stav používaných GIS systémov. Dôraz je kladený najmä na architektúru GIS systémov a používaných databáz.

2.1 Modelovanie reality z geografického hľadiska

V geografických informačných systémoch sa snažíme modelovať reálny svet, ktorý nás obklopuje. Realita je zložená z rôznych objektov a javov, ktoré môžeme popísať prostredníctvom ich vlastností. Človek ich dokáže vnímať svojimi zmyslami a rozoznať ich na základe ich špecifických vlastností, ich správania alebo ich interakcie k okolitým objektom.

Pojem objekt [11] chápeme z geografického hľadiska ako ľubovoľne objektívnu existujúcu vec (napr. cesta, les, budova, pôda, hory a množstvo iných), jav alebo proces odohrávajúci sa v reálnom svete. Môžeme hovoriť aj o výsledku určitého abstraktného uvažovania vyjadreného napr. pomocou rastrov – hustota obyvateľstva, zloženie obyvateľstva, miera nezamestnanosti atď. Objekt môžeme vyjadriť aj pomocou veličiny či už kvalitatívne alebo kvantitatívne. Podobne chápeme aj objekt v geoinformatike. Ide teda o akýkoľvek objekt záujmu, jav alebo veličinu. Presnejšie definovaný ako unikátny objekt, ktorý má presne určenú geografickú polohu v geopriestore a má definovanú geometriu, topológiu, tematické vlastnosti a dynamiku v čase. V geografických informačných systémoch sa vytvára modelový svet, ktorý predstavuje zjednodušený obraz reálneho sveta zachytávajúci jednotlivé geoobjekty. V procese modelovania geografickej reality pomocou geoinformačných systémov sa dôsledne rozlišuje [12]:

- reálny svet, ktorý modelujeme a v ktorom identifikujeme jednotlivé geografické objekty (skrátene geoobjekty a angl. objects) reality,
- modelový svet, ktorý zobrazuje len tie časti reálneho sveta, ktoré sú relevantné z hľadiska riešenej problematiky prostredníctvom geografických prvkov (skrátene geoprvkov a angl. features), ktoré predstavujú modely reálnych objektov (obr. 2.1).



Obr. 2.1: Abstrakcia modelovania reálneho sveta.

Modelovaný svet existuje len v prostredí GIS. Vychádzajúc z jednej jeho vlastnosti, GIS zobrazuje len podstatné zložky reálneho sveta a nepodstatné vynecháva. Určenie, čo je podstatné a čo nie, závisí od účelu, ktorý má daný GIS plniť. Napríklad pri modelovaní vodných tokov sú najpodstatnejšími geoobjektami rieky, pričom uvažovať iné objekty ako napr. lesy, pôdne typy atď. je nepodstatné.

Neustály dopyt po informáciách a poznatkoch získavaných pomocou GIS vzrastá. Výsledkom je rýchly proces štandardizácie geoinformačných pojmov, ktorý ústi do tvorby množstva medzinárodných noriem. Cieľom týchto noriem je sprístupniť tieto informácie rôznym aplikáciám, systémom, technológiám. To si vyžaduje normalizovaný spôsob ich definovania a opisu, normalizovanú metódu ich štruktúrovania a kódovania pre potreby aktualizácie, prenosu a sprístupnenia nezávisle od akéhokoľvek technicko-programového vybavenia.

2.1.1 Modelovanie geografických objektov - geoobjektov

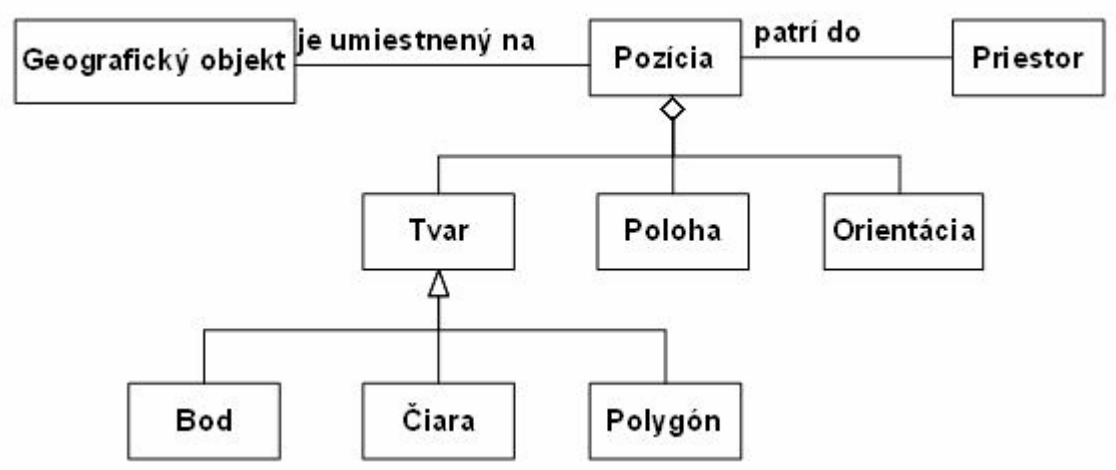
Pre pochopenie jednotlivých krokov modelovania geografickej reality v prostredí (geo)informačných systémov sú potrebné aspoň základné znalosti z teórie a praxe tvorby báz dát súhrnne označovaných ako dizajn báz dát [13] .

Modelovanie geoobjektov, javov a procesov v prostredí geografických informačných systémoch môžeme zosumarizovať do týchto základných fáz:

- špecifikácia objektov,
- špecifikácia predmetu modelovania,
- vytvorenie abstraktnej používateľsky orientovanej geografickej informačnej štruktúry, t.j. logický model dát a jeho štruktúra,
- zavedenie, resp. implementácia geografickej informačnej štruktúry do počítačového prostredia, t.j. fyzická štruktúra súborov dát.

Výsledkom konceptuálneho modelovania je konceptuálna schéma geografickej bázy dát (obr. 8).

Geografická báza dát obsahuje okrem iného aj popis obsahu geografických dát, štruktúry geografických dát a definované pravidlá tvorbu a manipuláciu s geografickými dátami. Ako je možné vidieť na obr. 2.2. pri istej miere abstrakcie uvažujeme základné triedy. Trieda geografický objekt obsahuje objekty, ktoré sú umiestnené v určitej pozícií – trieda pozícia – a patria do priestoru. Trieda pozícia agreguje triedy tvar, poloha a orientácia. Slúžia pre popis pozície daného objektu. Navyše triedy bod, čiara a polygón dedia metódy a funkcie od triedy tvar. Pri tvorbe koncepcie modelu musíme dbať na to, aby koncepcia modelu bola nezávislá od použitých technických a programových prostriedkov v procese štruktúrovania a zavedenia modelu dát do počítača.



Obr. 2.2: Konceptuálny model geoobjektov.

Po vytvorení konceptu modelu geografických dát môžeme uvažovať nad tvorbou účelového modelu bázy dát v programovom prostredí GIS systémov. Účelový model bázy dát tvorí aplikačnú vrstvu, ktorá nadväzuje na predošlý model. Pomocou aplikačnej vrstvy môžeme vytvárať a riadiť štandardizované informačné modely bázy dát, ktoré využívajú konkrétny logický model dát. Logický model dát môže mať formu relačného, objektového a iných modelov. Aplikácia, ktorá takto vznikne, má za cieľ vytvoriť a používať definovanú geografickú štruktúru pre dosiahnutie stanovených cieľov. V spomínanej fáze modelovania sa identifikujú dáta špecifikujúce geografické objekty, ich vlastnosti a vzťahy medzi objektmi, ktoré sú podstatné z hľadiska stanoveného účelu, využitím nástrojov vybraného modelu dát a jeho systému riadenia. Medzi vlastnosti, ktoré sa definujú, patria v prvom rade priestorové, tematické a časové vlastnosti objektov.

V ďalšej fáze nasleduje implementácia. Konkrétny geografická dátový model reality sa implementuje do „virtuálneho sveta“. Implementácia sa vykonáva na základe definovaného logického modelu dát, teda na základe aplikačnej schémy. Používateľ tým dostane možnosti a prostriedky na to, aby si mohol odvádzať potrebné geografické informácie. Momentálne je

zaužívaných niekoľko prístupov pre tvorbu dátového modelu. V praxi je zatiaľ najpoužívanejší entitno-relačný prístup, ako bude neskôr vidno aj z prípadovej štúdie. Tento prístup využíva pre tvorbu modelov entity. Tie sú chápané ako určitý model objektov reálneho sveta. Ich vlastnosti sú reprezentované pomocou atribútov a vzťahy medzi objektmi sú reprezentované reláciami. Entitno-relačný prístup sa implementuje do GIS systémov v prevažnej miere buď do samostatných relačných alebo objektovo-relačných modelov dát. Podrobnejšie si rozoberieme entitu ako model geografického objektu reálneho sveta.

Entita reprezentuje *unikátny objekt modelovania*. Entity je možné zoskupovať do tried na základe spoločných vlastností. V prípade, že hovoríme o jednej entite z triedy entít, ide o jej inštanciu. Definície vlastností entít sa označujú ako atribúty. Každá entita nadobúda jedinečné hodnoty atribútov. Rozsah hodnôt atribútov je daný oborom hodnôt. Z toho vyplýva, že atribút môže nadobúdať len hodnoty určitého typu a určitého rozsahu. Dve a viac entít môžu by byť navzájom prepojené rôznymi vzťahmi, tzv. reláciami. Podľa počtu entít, ktoré sú navzájom prepojené existujú viaceré možnosti relácií. Vzťahy 1:1, 1:N, N:N atď. Je nutné povedať, že tento prístup modelovania reality je momentálne najviac využívaný aj pri modelovaní GIS systémov. Ako príklad uvedieme modelovanie napr. lesného porastu. Základné entity, ktoré je možné definovať, sú entity strom, les, podnebie atď. Podobné entity sú zoskupované do tried. Za podobné entity môžeme považovať borovicový les, smrekový les, ktoré sú zoskupené do entity Ihličnaté lesy. Podobne tak môžu byť zoskupené listnaté lesy a tie spolu s ihličnatými lesmi môžu byť ešte zoskupené do entity les. Postupne sa tak prechádza až po najväčšiu mieru abstrakcie. Relácií medzi týmito entitami môže byť mnoho. V každom lese môže byť 1:N listnatých aj ihličnatých lesov, každý strom je priradený práve jednému stromu atď. Takýto spôsob modelovania GIS systémov je veľmi rozšírený a existuje množstvo štandardov, ktoré tento spôsob využívajú.

2.1.2 Geografické objekty a geografické prvky

Všetky objekty spolu s modelmi s podobnými vlastnosťami sa dajú zoskupovať v GIS systémoch do tried, ktoré definujú funkčnosť a štruktúru každej triedy podobných objektov. Každý špecifický individuálny objekt tvorí výskyt danej triedy, pričom *geografický objekt* alebo *geoobjekt* je reálny alebo abstraktný predmet modelovania.

Geografický prvok (*geoprvok)* [14] *je model* reálneho objektu. Každý geografický prvok zahrňuje priestorovú lokalizáciu, to znamená, že sa dá priestorovo zamerať na voliteľnú časť zemského povrchu. Existujú dva druhy geoprvkov, reálne a abstraktné. Preto rozlišujeme triedy zoskupujúce geoprvky na triedy, ktoré zoskupujú reálne geoobjekty a triedy, ktoré zoskupujú abstraktné geoobjekty. Prvou z nich je trieda reálnych geoobjektov.

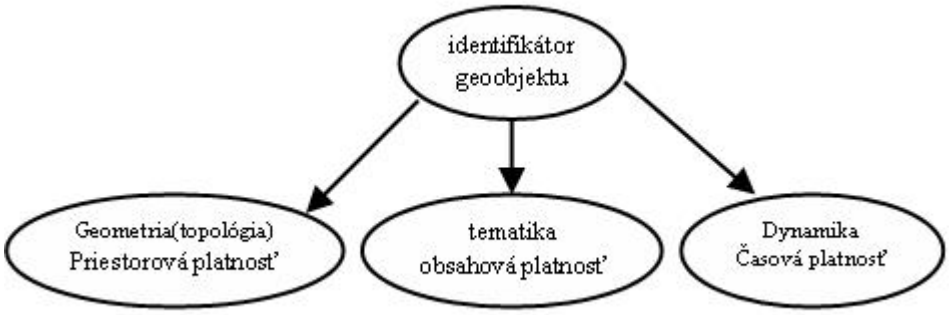
Pod geoprvkovou triedou reálnych objektov rozumieme napríklad všetky mestá, pričom každé konkrétne mesto predstavuje geoprvok, ktorý sa nedá ďalej členiť na ďalšie mestá. Na druhej strane sa ale dá rozdeliť na jednotlivé časti — parcely, budovy a pod. Druhým typom triedy je trieda geoprvkov abstraktných geoobjektov. Týmito objektmi sa chápu napr. administratívne hranice, vekové zloženie obyvateľstva, vrstevnice atď.

Zoskupovanie entít do tried predstavuje pri modelovaní veľkú výhodu. Všetky entity, ktoré spolu či už tematicky alebo logicky spolu súvisia, môžeme popísať formalizovaným spôsobom ako prvky objektových tried. Aj keď majú geoobjekty rôznu geometriu, môžu mať aspoň čiastočne podobné tematické vlastnosti, a preto ich je možné spracovávať podobnými, až rovnakými metódami. Fakt, že aj zdanlivo nesúvisiace geoobjekty je možné spracovávať podobnými metódami, ukážem na príklade. Analyzujem riečnu sieť v GIS systéme. Každá rieka spadajúca pod riečnu sieť má aspoň jeden začiatok a jeden koniec. Ďalej rieky majú určitú dĺžku, šírku, prietok alebo spád vody. Všetky tieto vlastností je možné zmerať v teréne pomocou špeciálnych nástrojov. Ak uvažujeme, že všetky rieky v danom geopriestore tvoria riečnu sieť, tak platí nasledovné. Ak je geoobjektom rieka Dunaj, ktorá patrí do riečnej siete, tak automaticky získava všetky vlastnosti triedy riečna sieť. Tieto vlastnosti sú spoločné pre všetky geoobjekty danej triedy. Samozrejme každý geoobjekt zapuzdruje aj svoje vlastné vlastnosti, dodatočné atribúty, ktoré sú typické len pre daný geoobjekt. Napríklad názov rieky, počet vodných elektrárni, rozsah znečistenia atď.

Ako opísať geoobjekt správne a čo najviac podobne realite závisí od viacerých faktorov. Každý geografický prvok, ktorý chceme správne reprezentovať v GIS systéme, musí byť charakterizovaný z viacerých hľadísk. Najdôležitejšie sú, už niekoľkokrát spomínané, poloha v priestore, geometrické vlastnosti, tematické vlastnosti, časový vplyv na geoobjekt a vzťahy medzi okolitými geoobjektami. Všeobecne povedané každý geoobjekt by mal byť definovaný minimálne svojou polohou, ktoré môže byť absolútna a relatívna, tematikou a dynamikou.
Význam jednotlivých vlastností je nasledovný:

- *Absolútna poloha* – geometria - popisuje priestorovú lokalizáciou geoprvku v príslušnom geografickom priestor. Absolútna poloha býva vyjadrená pomocou geometrických kompozít ako napr. bod, čiara, polygón a pod.,
- *Relatívna poloha* – topológia - opisuje vzájomné priestorové vzťahy (relácie) s inými geoobjektmi (susednosť, prekrytie, orientáciu, atď.),
- *Tematika* je určená atribútmi, ktoré opisujú tematické charakteristiky geoobjektu (typ triedy, podtrieda, špecifické vlastnosti atď.),
- *Dynamika* vyjadruje časové charakteristiky, ktoré opisujú zmeny geometrie, topológie a atribútov geoobjektu v čase.

Identifikácia geoobjektu v rámci GIS systému je realizovaná prostredníctvom jedinečného identifikátora, ktorý na geoobjekt ukazuje. Identifikátor každý objekt identifikuje od ostatných geoobjektov a zjednocuje jeho vlastnosti v príslušnom dátovom modeli. Takýmto identifikátorom môže byť v GIS systémoch napr. číslo parcely, názov obce, adresa alebo hocijaký iný typ identifikátora, ktorý je samozrejme vhodne vybraný v závislosti od koncepcie dátového modelovania. Základné charakteristiky geoprvku sú znázornené na obr. 2.3.



Obr. 2.3: Základné charakteristiky geoprvku .

Modelovanie geoobjektov na rôznych úrovniach abstrakcie geografického priestoru je možné realizovať napr. na úrovni veľkého územia (štátu), na úrovni kraja, okresu alebo obce. Všetky geoobjekty sa môžu vzťahovať k rôznym referenčným systémom. Dvojrozmerný geografický model, známy ako mapa, najčastejšie slúži na prezentovanie geoobjektu umiestneného na zemskom povrchu. Spomínaný model môže byť reprezentovaný rôznymi súradnicovými systémami. Samotné GIS systémy pracujú so zjednodušenými modelmi reality, nie so skutočnými objektmi. Takéto modelovanie má viacnásobné použitie v oblasti vytvárania geoobjektov, ich geometrie, tematiky a dynamiky. Pre ten istý geoobjekt môžu byť dôležité rôzne charakteristiky v závislosti od definovaného účelu.

Pri modelovaní geoobjektov využívame na ich popis geografické informácie a geografické dáta. ***Geografická informácia*** je informácia o určitom mieste nachádzajúcom sa na zemskom povrchu. Vyjadruje znalosť *kde* sa niečo (geoobjekt) nachádza, resp. čo sa nachádza na určitej polohe. Geografické informácie charakterizujú geoobjekty s dôrazom na príslušné faktory, ktoré sú dôležité z hľadiska ich aplikácie v praxi. Stávajú sa významným prostriedkom poznania geografického priestoru, ak s nimi pracujeme ako s poznatkami o geoobjektoch reálneho sveta.
Jedným zo spôsobov ako vytvoriť geografické informácie sú informačné technológie. Pomocou nich sa dajú uložiť, vybrať a vizualizovať potrebné dáta. Pre distribúciu geografických informácie sú počítačové siete ideálnym prostriedkom. Znamená to, že dostupnosť týchto informácií je

takpovediac z každého miesta na zemi. Získať informácie zo zdroja a pracovať s nimi je jednou z čŕt GIS systémov a týmto riešením je dostatočne pokrytá. Technológie geografických informácií sú použiteľné pri akýchkoľvek rozhodnutiach s priestorovými súvislosťami, ktoré je potrebné prijať. Musia byť však prispôsobené konkrétnej situácii.

Druhým typom dát slúžiacich na popis geoobjektov sú *geografické dáta*, tzv. geodáta. Sú to informácie týkajúce sa javov alebo geoobjektov, ktoré majú svoju polohu priamo priradenú k Zemi vo forme počítačovo spracovateľnej. Ich úlohou je charakterizovať geografickú polohu, vlastnosti javov, geoobjektov a vzťahov medzi nimi, prezentujú abstrakcie objektov reálneho sveta. Formálne ide o opisy geografických informácií vo forme čísiel a znakov vhodných na počítačové spracovanie, t. j. digitálne dáta, ktoré oproti analógovým sa dajú prenášať elektronickými zariadeniami [15].

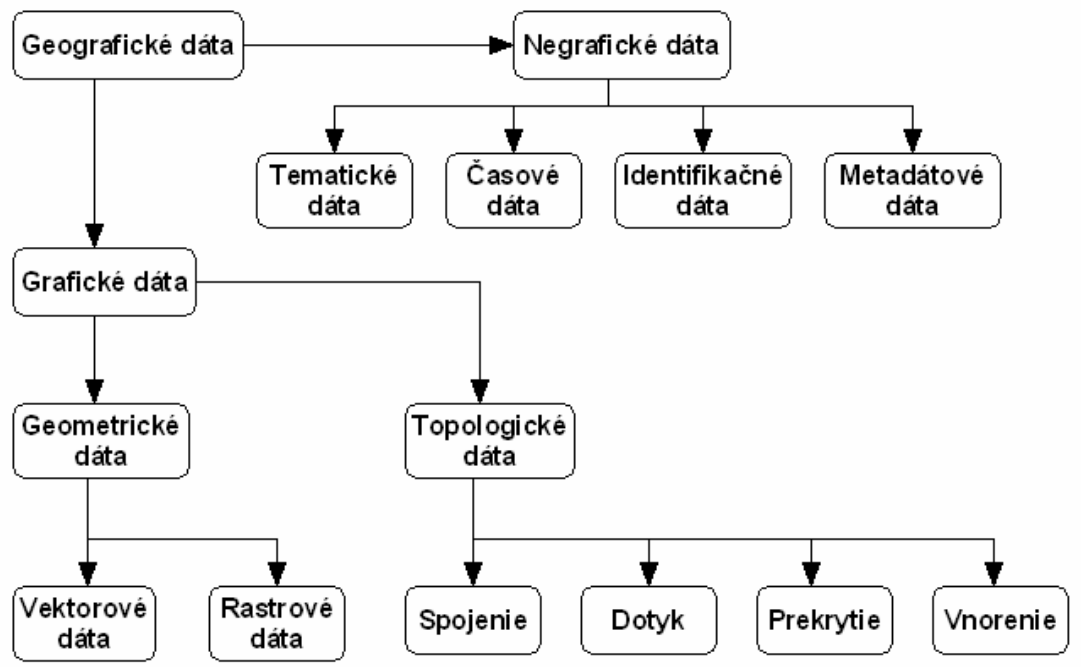
Geografické dáta opisujú všetky vlastnosti geoobjektov - polohu, tvar a relácie medzi geoobjektami - väčšinou vo forme geografických súradníc a topológie. Tieto dáta sú potrebné na priestorovo orientovanú poznávaciu a rozhodovaciu činnosť rôznych štátnych, samosprávnych, súkromných a iných subjektov. V tomto zmysle sú geodáta využívane napr. aj v GIS systémoch pre podporu rozhodovania práve na základe povahy a vlastnosti geodát. Za ich základné vlastnosti sa považuje komplikovaná tematická, územná a časová štruktúrovanosť. *Geografické dáta* sa, na rozdiel od iných typov dát, skladajú minimálne z dvoch základných zložiek:

- *priestorovo-identifikačnej* najčastejšie vo forme geografického identifikátora, ktorý identifikuje priestorovú lokalizáciu, vzťahy a zabezpečuje väzbu na konkrétne miesto v priestore,
- *tematickej, známej aj ako atribútovej* vyjadrujúcej kvalitatívnu alebo kvantitatívnu charakteristiku ukazovateľa.

2.1.3 Geografické prvky a ich vlastnosti

Geografické prvky by sme mohli definovať nasledovne. Geoprvky sú opísané pomocou geografických informácií na báze geografických dát. V kartograficky orientovaných GIS sa tieto dáta zvyknú hierarchicky členiť podľa schémy na obr. 2.4. Tato hierarchia a filozofia členenia geoprvkov najlepšie zodpovedá klasickej mapy.

Geografické dáta sa na najvyššej úrovni delia na grafické a negrafické. Negrafické dáta opisujú vlastnosti geoobjektov, ktoré sa netýkajú ich priestorovej orientácie. Na druhej strane grafické dáta popisujú graficky geometriu geoobjektov, ich topológiu. Negrafické dáta definujú ďalšie vlastnosti geoprvkov — tematické, časové, identifikačné, metadátové a ďalšie. Medzi tento typ dát môžeme zahrnúť aj dáta opisujúce operácie, ktoré možno s príslušným geoobjektom vykonávať



Obr. 2.4: Členenie geografických dát v GIS.

Geometrický popis geoprvkov je založený na dvoch základných geometrických modeloch, a to na priestorovo spojitom *implicitnom* a diskrétnom *explicitnom* modeli. Implicitný (absolútny) model využíva funkčné polia (napr. matematické funkcie) na modelovanie výskytu (polohy) modelovaných reálnych objektov, ktoré majú spojitý charakter výskytu. V GIS systémoch sa tento model vytvára v pomocou rastrových, resp. mozaikových priestorových štruktúr. Explicitný (relatívny) model využíva objektový prístup modelovania reality. Realita je modelovaná pomocou jednoznačne identifikovateľných (diskrétnych) prvkov. V GIS systémoch sa tieto modely realizujú vo forme vektorových priestorových štruktúr.

Rozšírená charakteristika alebo opis geografické prvku dátami sa dá rozdeliť do šiestich základných zložiek [16]:

- *geometrickej*, ktorá zaznamenáva lokalizáciu geoobjektu v priestore, jeho geometrické charakteristiky a priestorové vzťahy s inými objektmi,
- *atribútovej* popisujúcej negeometrické tematické (obsahové) charakteristiky geoobjektu,
- *časovej*, ktorá referuje o dobe existencie geoobjektu pri danom stave geometrie a atribútov,
- *vzťahovej* popisujúcej vzťahy, do ktorých geoprvok vstupuje s inými prvkami,
- *funkčnej*, ktorá popisuje operácie, ktoré sa dajú prevádzať s geoobjektu,

- *kvalitatívnej*, ktorá tvorí doplňujúcu zložku a vzťahuje sa k celkovej charakteristike geoobjektu pričom referuje o jeho kvalite, akosti.

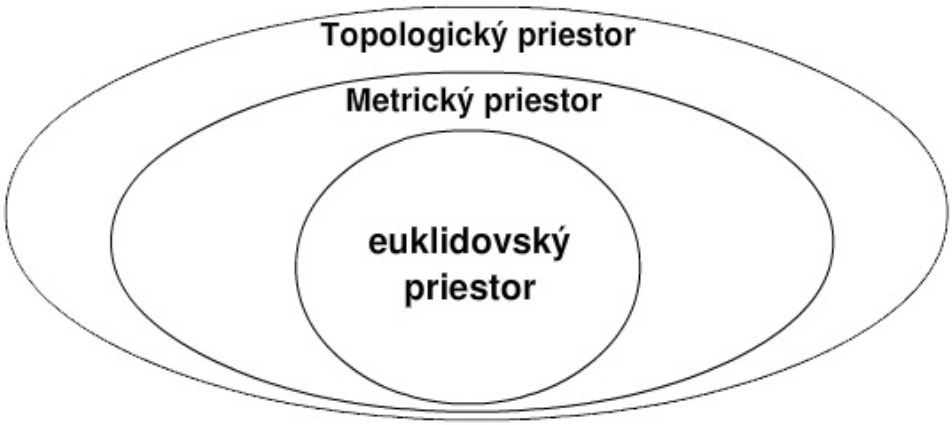
Každý geoobjekt má svoju geometriu. Pod geometriou geoobjektu rozumieme všetky informácie vzťahujúce sa k polohe a rozmerom modelovaných geoobjektov v priestorovom referenčnom systéme. Geometrická zložka popisu geoobjektov sa viaže na:

- *definovanie geografického priestoru* — dimenzie priestoru a metriky, v ktorom sa geoprvok lokalizuje a merajú vzdialenosti,
- *stanovenie priestorovej polohy* geoobjektu v definovanom geometrickom priestore,
- *definovanie topológie*, t. j. určenia priestorových vzťahov geoobjektu s inými geoobjektami v geografickom priestore,
- *definovanie priestorových vlastností* jednotlivých geoprvkov a ich tried.

2.1.4 Umiestnenie geoobjektov v priestore

K modelovaniu v GIS systémoch patrí aj definovanie priestoru, v ktorom budú zobrazované modelované geoobjekty. Spravidla sa priestor definuje podľa toho akému účelu má daný model slúžiť. Najčastejšie sa používa euklidovský priestor, ktorý je vhodný pre väčšinu aplikácií, napr. v katastri nehnuteľností, pozemkovej správe atď. Ďalším priestorom, ktorý sa používa, je topologický priestor. Ten je vhodný pre účely, kedy chceme zistiť napr. najkratšiu cestu z jedného miesta do druhého. Podstatou v tomto priestore je graf, ktorý modeluje tvar a vzdialenosť určitých cestných spojení v časových alebo napr. cenových jednotkách.

Existuje viacero definovaní priestorov a ich hierarchické usporiadanie. Vzhľadom k tomu, že nie všetky priamo súvisia s modelovaním geoobjektov, si uvedieme len niektoré, ktoré sú znázornené na obr. 2.5.



Obr. 2.5: Hierarchické rozdelenie priestorov.

Medzi priestory, ktoré sa ako prvé približujú k ľudskému vnímaniu priestoru, patria topologické priestory. Tie sú tvorené algebraickými a kombinatorickými grafmi. V topologických priestoroch sa už objavuje pojem okolie. Ak zavedieme do tohto priestoru vzdialenosť, dostaneme metrický priestor. A keď k tomu pridáme aj smer, dostaneme euklidovský priestor. Keďže tento typ priestoru sa považuje za najčastejšie používaný, venujeme mu najväčšiu pozornosť. V GIS systémoch sa samotný pojem priestor chápe ako množina objektov, ktorá má určité vlastnosti podobné fyzikálnemu priestoru v realite. Geometria prvkov je opísaná pomocou súradnicového systému, ktorý má svoju dimenziu. V prípade euklidovského priestoru musia platiť nasledovné podmienky:

- identita, matematicky vyjadrená ako: $d(A,B) \geq 0$ a $d(A,B)=0$ len ak A=B, t.j. vzdialenosť dvoch bodov je vždy nenulová, pokiaľ nejde o totožný bod,
- symetria : $d(A,B) = d(B,A)$, t.j. vzdialenosť z bodu A do bodu B je rovnaká ako vzdialenosť z bodu B do bodu A,
- trojuholníková nerovnosť: $d(A,B) \leq d(A,C) + d(C,B)$.

Pri modelovaní geoprvkov môžeme uvažovať rôzne dimenzie priestoru. Čím je však dimenzia priestoru vyššia, tým je potrebné uvažovať všetky nezávislé smery v priestore, ktoré sú potrebné na jeho opis, a tým sa modelovanie výrazne komplikuje. V prípade, že máme k dispozíciu topologickú informáciu „bod A je blízko bodu B“ a „bod C je blízko bodu B“, nemôžeme potvrdiť identickosť objektov A a C. Tým pádom nemôžeme použiť túto informáciu ani na výpočty typu aká je vzdialenosť medzi týmito bodmi. V súčasnosti sa geoobjekty modelujú najčastejšie v 2-rozmernom priestore. Matematicky vzaté ide o rovinu, kde každý geoobjekt má definovanú polohu a geometrický tvar pomocou súradníc x a y. Okrem 2-dimenzionálneho priestoru sa ešte geobjekty modelujú aj v tzv. 2,5-dimenzionálnom priestore. Je to priestor, kde sa súradnica x nechápe ako priestorová súradnica ale ako tematicky atribút. Najmenej používané priestory sú 3-dimenzionálne a 4-dimenzionálne. V 3D priestore sa uvažuje práve súradnica x ako tretia priestorová súradnica. V 4D priestore sa k tomu pridáva ešte časová zložka a modelujú sa geoobjekty meniace sa v časovom horizonte. Z hľadiska *geometrického modelovania* v GIS sa rozlišujú tieto typy geoprvkov:

- *0-dimenzionálne*, t. j. bezrozmerné *body*, ktoré nemajú dĺžku, smer ani plochu (napr. výšková kóta, centroid sídla atď.),
- *1-dimenzionálne*, t. j. *čiary* charakterizované dĺžkou a smerom, ktoré nemajú definovanú plochu (napr. cestný úsek, smer migrácie),
- *2-dimenzionálne*, t. j. *areály*, ktoré majú plochu a dĺžku (napr. parcela, pôdorys mesta, hon),
- *3-dimenzionálne*, t. j. *telesá* s objemom (napr. vodná nádrž, budova), resp. *mnohosteny*

ohraničené rovinami, ale bez obsahu (napr. budova ako konštrukčné teleso),

- *4-dimenzionálne*, t. j. *animácie*, ktoré zaznamenávajú časové zmeny (dynamiku) predchádzajúcich dimenzií (1-3D), kde čas sa chápe ako štvrtá dimenzia (napr. animácia pohybu vozidiel v meste, dynamiky vývoja osídlenia).

Dimenzia modelovaných geoobjektov sa môže meniť v závislosti od toho v akej miere abstrakcie daný problém riešime. Podobne je možné dimenzionalitu aplikovať aj na definovanie topológie geoobjektov. Najčastejšie ide o tieto typy uzly — *0D*, hrany — *1D*, polygóny — *2D* a telesá/mnohosteny — *3D*. Dimenzionalita v tomto zmysle sa taktiež uplatňuje aj pri tematickom a temporálnom modelovaní, kde množina charakteristík geoobjektu prezentuje tematickú alebo časovú dimenziu. Na záver spomenieme, že všetky geoprvky sa dajú modelovať v rôznych mierkach, resp. v rôznom priestorovom rozlíšení. Poznáme 3 základné typy mierky:

- veľká mierka (makromodelovanie),
- stredná mierka (mezomodelovanie),
- malá mierka (mikromodelovanie).

V GIS sa najčastejšie používa rôzna absolútna mierka. Hodnoty atribútov geoprvkov v jednotlivých triedach sa môžu meniť v závislosti od ich polohy. Sú v podstate priestorovou premennou indikujúcou priestorový proces napriek tomu, že sa v GIS explicitne neuvažuje o ich zmene v čase. Analýza a modelovanie priestorových štruktúr a procesov sú základnými otázkami GIS systémov. Bez ohľadu na tematickú stránku jednotlivých riešených problémov pomocou nástrojov GIS sú v nich implementované podobné metodologické postupy opisu priestorovej zmeny vlastností modelovaných geoobjektov. Ide najmä o opisy pomocou:

- *priestorového rozdelenia*, t. j. určenia priestorovej variability procesov, resp. ich korelácie s inými procesmi,
- *priestorového priemerovania*, t. j. výpočtu priemerných hodnôt (váh) založeného na bodovo lokalizovaných dátach,
- *priestorového rozlíšenia*, t. j. konštrukcie homogénnych regiónov (areálov) s využitím princípu minimálnej priestorovej variácie v regióne a maximálnej variácie medzi dvoma rôznymi regiónmi (typizácia a regionalizácia),
- *priestorovej interpolácie a extrapolácie*, t. j. detailného opisu, resp. generalizácie pomocou funkcionálnych polí a povrchov,
- *zmeny priestorovej mierky*, t. j. zväčšovania (priestorovej agregácie) a zmenšovania (disagregácia) prostredníctvom generalizácie geometrie, topológie alebo tematického obsahu

geoprvkov.

Geovedné problémy sú nie vždy chápané a skúmané ako priestorové problémy, pri riešení ktorých sa dajú využiť priestorovo-časové analytické metódy a modelačné techniky, ktoré sa postupne implementujú do technológií GIS a overujú v praxi.

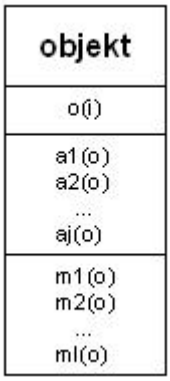
3 Návrh objektovo-orientovaného GIS systému

Po dôkladnom zanalyzovaní všetkých vlastností objektovo-orientovaných databáz bude navrhnutý prototyp návrhu geografickej objektovo-orientovanej databázy. Bude vychádzať zo základných čŕt týchto databáz obohatených o geografické informácie vhodné pre ukladanie geodát. Návrh bude popísaný teoreticky a podložený vhodnými obrázkami pre lepšie pochopenie.
Každý objekt vyskytujúci sa v databáze musí spĺňať nasledujúce podmienky:

- objekt musí byť jednoznačne určený unikátnym identifikátorom *i*,
- objekt zapuzdruje neprázdnu množinu atribútov $A = (a_1, a_2, ... ,a_j)$, kde *j* = celkovému počtu atribútov objektu,
- hodnoty atribútov prislúchajúce k jednotlivým atribútom objektu tvoria množinu $H = (h_1, h_2, ... ,h_k)$, kde *k* = celkovému počtu atribútov objektu,
- objekt obsahuje neprázdnu množinu metód $M = (m_1, m_2, ... ,m_l)$, kde $l >= 1$.

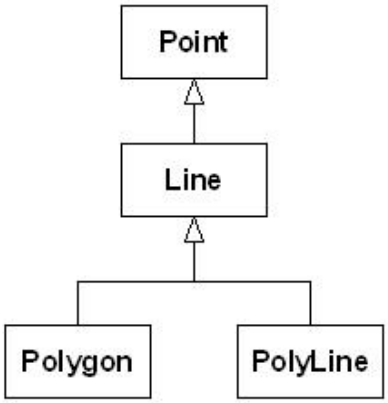
Je nutné podotknúť, že hodnotami atribútov môžu byť jednak štandardné hodnoty typu integer, string, byte atď., ale aj referencie na iné objekty v databáze.

Základná schéma objektu v databáze, z ktorej budeme vychádzať je uvedená na obr. 3.1, kde o(i) je všeobecný identifikátor objektu, a_1 až a_j sú atribúty objektu nadobúdajúce hodnoty h_1 až h_k a metódy prislúchajúce k tomuto objektu sú m_1 až m_l. Do atribútov zložky patria dve skupiny atribútov. Sú to atribúty, ktoré sú implicitne nadefinované a týkajú všeobecných vlastností objektu a atribúty užívateľské, ktoré si môže používateľ sám dodefinovať na základe rozšírených požiadavok.



Obr. 3.1: Základná schéma objektu v databáze.

Z obr. 3.1 budeme vychádzať pri reprezentovaní všetkých vektorových geoprvkov v databáze. Postupne si navrhneme schému objektov pre body, línie, zložené línie a polygóny. Objekty budú zaradené do hierarchie od najjednoduchších až po najzložitejšie. Hierarchia tried vektorových objektov je uvedená na obr. 3.2. Najzákladnejšou triedou je trieda Point. Tá reprezentuje samotný bod, ktorý je súčasťou všetkých ostatných vektorových objektov. Slúži najmä na základné identifikovanie geoobjektov, ich zemepisných súradníc. Od tejto triedy je odvodená trieda Line. Každá línia je reprezentovaná práve dvoma bodmi. S malými odlišnosťami je od tejto triedy odvodená trieda PolyLine, čo je reťazec línií. Oveľa komplexnejšou triedou, ktorá je takisto odvodená od triedy Line, je trieda Polygon. Ako je možné vidieť všetky triedy vzchádzajú z najjednoduchšej triedy reprezentujúcej objekt typu bod.



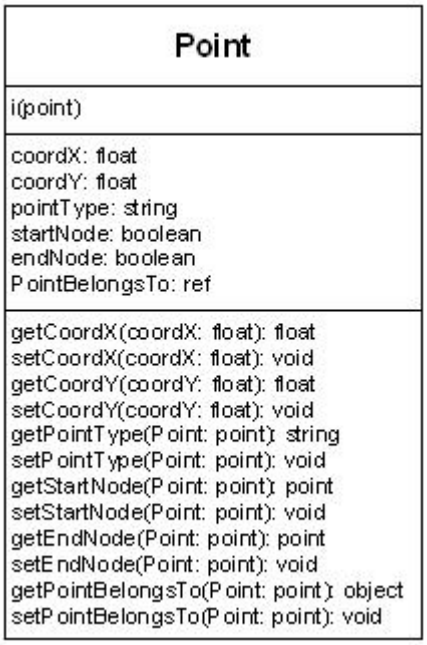
Obr. 3.2: Hierarchia tried vektorových objektov.

3.1 Návrh ukladania vektorových objektov

Po oboznámení sa s hierarchiou tried geoobjektov a ich zložitosťou si rozoberieme všetky vektorové geoobjekty. Pre oboznámenie notácie používaných diagramov je potrebné uviesť, že atribúty typu *ref* sú atribúty, ktorých hodnota je referencia na iný objekt. To isté platí pre metódy. Metódy s uvedeným návratovým typom *object* vracajú buď jednu alebo viacero referencii na iné objekty.

Prvým objektom, ktorý si rozoberieme, je vektor typu bod. Jeho reprezentácia je uvedená na obr. 3.3. Bod je inštancia triedy Point. Trieda obsahuje atribúty coordX a coordY určujúce súradnice bodu v súradnicovom systéme. Atribút pointType určuje, či ide o izolovaný bod, uzol alebo bod povrchu. V prípade, že ide o bod povrchu, nastaví sa skúmaná veličina a jej hodnota. V prípade, že daný bod je uzol, tak sa nastaví buď atribút startNode alebo endNode na true, resp. false, podľa toho či ide o začiatočný alebo koncový uzol. Ďalej každý uzol si uchováva informáciu o tom, ku ktorému zložitejšiemu vektoru patrí vo atribúte PointBelongsTo. Medzi metódy patriace do triedy Point patria:

- getCoordX, setCoordX – pre nastavenie, resp. získanie súradnice X,
- getCoordY, setCoordY – pre nastavenie, resp. získanie súradnice Y,
- getPointType, setPointType – pre nastavenie, resp. získanie typu bodu,
- getStartNode, setStartNode – pre nastavenie, resp. získanie začiatočného uzla,
- getEndNode, setEndNode – pre nastavenie, resp. získanie koncového uzla,
- getPointBelongsTo, setPointBelongsTo – pre nastavenie, resp. získanie všetkých vektorov, ku ktorým daný uzol patrí.

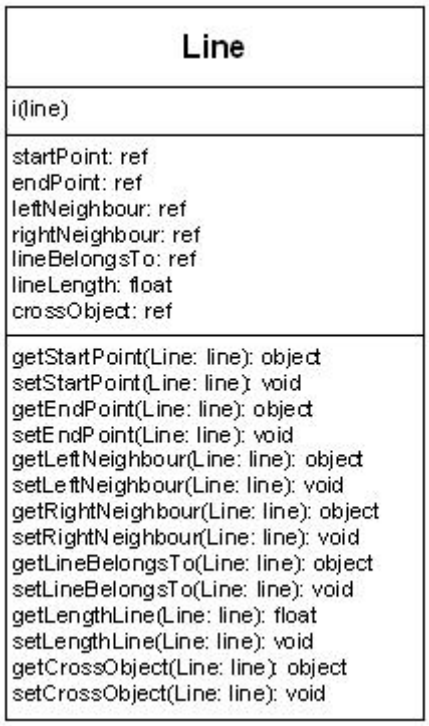


Obr. 3.3: Reprezentácia objektu bodu v databáze.

Druhým vektorovým objektom je línia uvedená na obr. 3.4. Línia je inštancia triedy Line, ktorá okrem svojich atribútov obsahuje referencie na uzly, ktoré ju tvoria. Táto trieda dedí od triedy Point všetky metódy týkajúce sa uzlov. Do metódy setLineBelongsTo sa budú ukladať vektory, ku ktorým daná línia patrí. Môže ísť buď o reťazce línií alebo polygónov. Ak to bude samostatná línia, tak jeho hodnota bude null. Tým, že poznáme u každej línií jej začiatočný – startPoint - a koncový bod - endPoint, vieme určiť jej orientovanosť. Na základe toho sú definované pre triedu Line atribúty LeftNeighbour – ľavý sused línie, RightNeighbour – pravý sused línie. Charakteristickými metódami je pre líniu aj LengthLine, ktorou je možné získať dĺžku danej línie a CrossObject, ktorou je možné získať všetky objekty pretínajúce danú líniu. Okrem spomínaných zdedených metód z triedy Point bude trieda Line obsahovať metódy:

- getStartPoint, setStartPoint – pre nastavenie, resp. získanie prvého bodu línie,

- getEndPoint, setEndPoint – pre nastavenie, resp. získanie druhého bodu línie,
- getLeftNeighbour, setLeftNeighbour – pre nastavenie, resp. získanie ľavého suseda,
- getRightNeighbour, setRightNeighbour – pre nastavenie, resp. získanie pravého suseda,
- getLineBelongsTo, setLineBelongsTo – pre nastavenie, resp. získanie objektov, ku ktorým línia patrí,
- getLengthLine, setLineLength – pre nastavenie, resp. získanie dĺžky línie,
- getCrossObject, setCrossObject – pre nastavenie, resp. získanie všetkých objektov, ktoré líniu pretínajú.

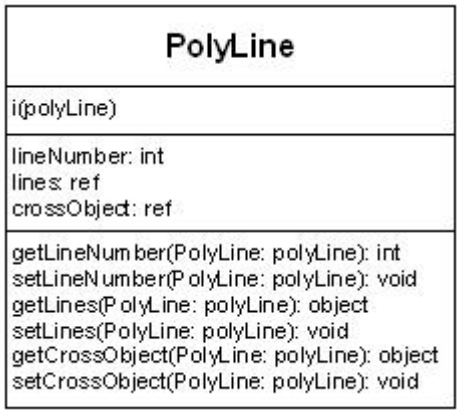


Obr. 3.4: Reprezentácia objektu línie v databáze.

Tretím objektom je reťazec línií znázornený na obr. 3.5. Reťazec línií je inštancia triedy PolyLine obsahujúca referencie na všetky línie, z ktorých je reťazec línií zložený, prostredníctvom atribútu lines. Táto trieda dedí metódy triedy Line. Navyše obsahuje atribút lineNumber, ktorý určuje z koľkých línií sa daný reťazec skladá a atribút crossObject uchovávajúci referencie na objekty pretínajúce reťazec línií. Preto sú dodefinované aj metódy:

- getLineNumber, setLineNumber – pre nastavenie, resp. získanie počtu línií v reťazci línií,
- getLines, setLines – pre nastavenie, resp. získanie línií, z ktorých je reťazec línií vytvorený,

- getCrossObject, setCrossObject – pre nastavenie, resp. získanie všetkých objektov, ktoré reťazec línií pretínajú.



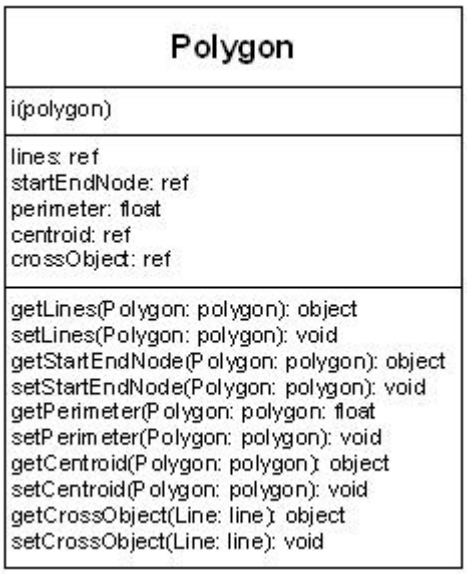
Obr. 3.5: Reprezentácia objektu reťazca línií v databáze.

Posledným objektom reprezentujúcim vektor je polygón uvedený na obr. 3.6. Polygón je inštancia triedy Polygon. Každá inštancia – objekt typu polygón - obsahuje referencie na všetky línií, z ktorých je zložený. Tie sú uložené v atribúte lines. Trieda polygón dedí metódy triedy Line. Navyše obsahuje atribút startEndNode, ktorý určuje zároveň začiatočný aj koncový uzol polygónu. Pre každý objekt tejto triedy vieme, z ktorých línií je zložený, s akými vektormi susedí a aj jednotlivé súradnice uzlov tvoriacich daný polygón. Každý polygón má uloženú informáciu o svojom obvode v atribúte perimeter. Atribút centroid slúži na uchovávanie referencie na bod nazývaný centroid. Centroid je ťažisko polygónu, ktorý sa v priestorových funkciách veľmi často využíva. Príkladom môže byť určenie zemepisných súradníc obce alebo mesta. Centroid sa berie ako abstrakcia mesta a reprezentuje jeden bod v rámci celého polygónu. V tejto triede pribudli nasledovné metódy:

- getStartEndNode, setStartEndNode – pre nastavenie, resp. získanie uzla, v ktorom polygón začína aj končí,
- getLines, setLines - pre nastavenie, resp. získanie všetkých línií, ktoré tvoria daný polygón,
- getPerimeter, setPerimeter - pre nastavenie, resp. získanie obvodu polygónu,
- getCentroid, setCentroid - pre nastavenie, resp. získanie centroidu polygónu,
- getCrossObject, setCrossObject - pre nastavenie, resp. získanie všetkých objektov, ktoré pretínajú daný polygón.

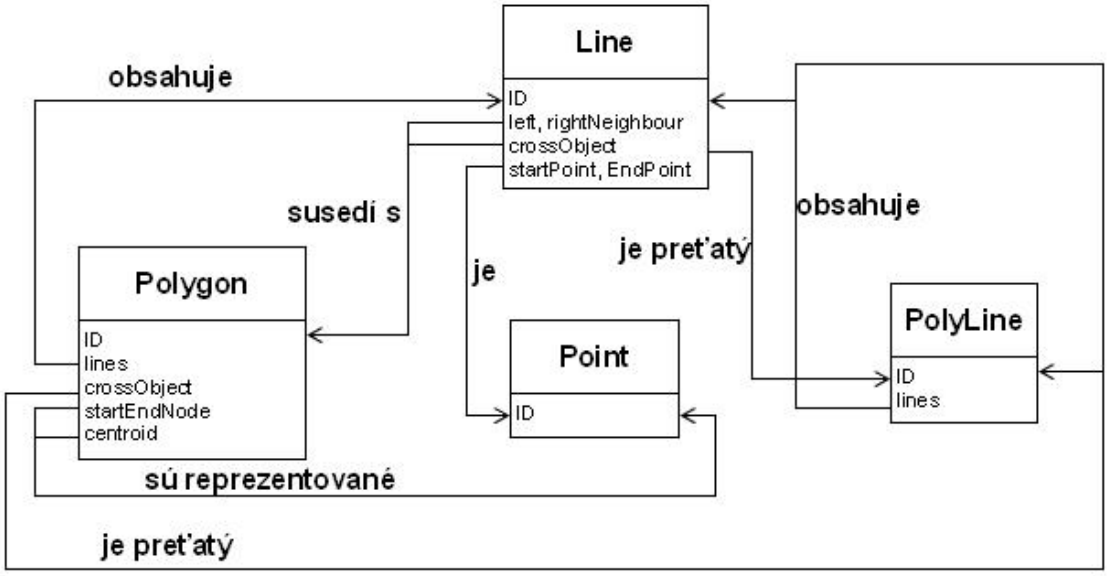
Všetky základné vektorové prvky sú popísané pomocou objektov s príslušnými atribútmi a metódami. Predpokladá sa, že používateľ si bude chcieť nadefinovať aj vlastné atribúty pre rôzne typy vektorových objektov. K tomu slúži metóda, ktorá je spoločná pre všetky triedy. Je to metóda

setUserAttribute, resp. getUserAttribute. Metóda obsahuje vstupné parametre, konkrétne názov nového atribútu, typ atribútu a jeho hodnotu. Výsledkom je priradenie nového atribútu aj s jeho hodnotou príslušnému vektorovému objektu. Metódy umožňujúce vytvorenie užívateľského atribútu sú súčasťou triedy Point, od ktorej dedia všetky ostatné triedy.



Obr. 3.6: Reprezentácia objektu polygón v databáze.

Na záver reprezentácie vektorov pomocou objektov si ešte popíšeme obrázok vyjadrujúci referencie objektov v objektovo-orientovanej databáze na ostatné objekty. Účel obr. 3.7 je načrtnúť dôležité vzťahy medzi objektmi.



Obr. 3.7: Vzťahy medzi objektmi v databáze.

Pre prehľadnosť sú na obr. 23. uvedené len niektoré atribúty tried geoobjektov, ktoré postačujú na ich identifikáciu. Názvy vzťahov medzi objektmi sú uvedené nad šípkou. Opäť môžeme vidieť, že najviac referencií na objekty obsahujú triedy Polygon a Line, keďže patria medzi triedy odvodené od jednoduchších tried. Polygón má svojich atribútoch uložené referencie na:

- všetky línie, ktoré ho tvoria,
- všetky objekty(línie, polygóny), ktoré ho pretínajú,
- bod, ktorý je spoločný pre jeho začiatok a koniec,
- centroid(bod).

Línia má vo svojich atribútoch uložené referencie:

- začiatočný a koncový bod,
- ľavého a pravého suseda(polygón),
- všetky objekty(línie, polygóny), ktoré ju pretínajú.

Považujem za dôležité spomenúť, že toto je jedna z veľkých výhod objektovo-orientovanej databázy v porovnaní s relačnou. Referencie na iné objekty úzko späté s daným objektom sú súčasťou daného objektu. Sú uložené priamo v ňom. Nie sú potrebné prístupy do rôznych tabuliek, ktoré by mohli byť tematicky príbuzné. Všetky informácie je možné získať prostredníctvom skúmaného objektu.
Všetky definované vektorové objekty komunikujú zasielaním správ. Každá správa pozostáva z:

1. názvu správy,
2. niekoľkých objektov, resp. identifikátorov objektov, nazývaných aj parametre správy,
3. odpovede.

Objekty majú definované rozhranie, ktoré určuje aké metódy sú prípustné pre daný objekt. Metóda je vyvolaná správou rovnomenného mena. Z toho vyplýva, že napr. pre objekt typu polygón sú dostupné len metódy triedy polygón atď. V prípade, ak objekt dostane správu, ktorá nezodpovedá žiadnej metóde, je automaticky vyslaná chybová správa naspäť objektu, ktorý správu poslal. Spomínané objekty môžu zaslaním správy iným objektom vyvolať tieto akcie:

1. pristúpiť priamo k atribútu, ktorý patrí danému objektu – načítať alebo zmeniť jeho hodnotu,
2. spustiť inú metódy prislúchajúcu danému objektu,
3. poslať správu inému objektu,
4. vytvoriť nový objekt.

3.2 Návrh ukladania rastrových objektov

Vzhľadom na štruktúru rastrov v GIS a jednotlivé typy rastrov som navrhol dva spôsoby ich uloženia v objektovo orientovanej databáze. Prvým spôsobom je definovanie rastrov a rastrových buniek ako samostatných tried. Navrhnutá definícia obsahovala triedu RasterCell reprezentujúcu jednotlivé bunky rastra. Keďže bunky rastra môžu mať rôzny tvar, či už pravidelný štvorcový, trojuholníkový alebo hexagonálny, musí to byť v tejto triede zohľadnené. Trieda RasterCell obsahuje tieto atribúty:

- cellType – určuje o aký typ rastrovej bunky ide,
- cellValue – určuje akú hodnotu má daná rastrová bunka,
- cellNeighbours – určuje susedov danej rastovej bunky.

Pre tieto atribúty je možné použiť nasledujúce metódy triedy RasterCell:

- getCellType, setCellType – pre nastavenie, resp. získanie typu rastrovej bunky,
- getCellValue, setCellValue – pre nastavenie, resp. získanie hodnoty rastrovej bunky,
- getCellNeighbours, setCellNeighbours – pre nastavenie, resp. získanie susedov rastrovej bunky.

Ďalšou triedou týkajúcou sa rastrov bude rovnomenné trieda Raster. Každá inštancia triedy Raster bude predstavovať určitý raster obsahujúci referencie na všetky svoje rastrové bunky. Ich počet bude mať uložený v atribúte cellsNumber. Pri tomto spôsobe ukladania rastrov do objektovo orientovanej databázy je očakávaný veľmi veľký počet objektov. Každý objekt typu Raster bude obsahovať milióny až stovky miliónov referencií na objekty typu RasterCell. Preto je jednou z požiadaviek objektovo orientovaná databáza, ktorá zvládne takéto množstvo objektov. Nakoniec dodefinujeme niektoré atribúty patriace triede Raster:

- cellsNumber – počet buniek prislúchajúci k rastru,
- rasterCells – referencie na všetky bunky prislúchajúce k rastru,
- rasterName – určuje názov rastru,
- rasterLegend – priraďuje textový popis k číselnému atribútu bunky.

Pre tieto atribúty je možné použiť nasledujúce metódy triedy RasterCell:

- getCellsNumber, setCellsNumber - pre nastavenie, resp. získanie počtu rastrových buniek v rastry,

- getRasterName, setRasterName - pre nastavenie, resp. získanie názvu rastra,
- getRasterLegend, setRasterLegend - pre nastavenie, resp. získanie textovej reprezentácie číselnej hodnoty rastrovej bunky.

Neskoršími úvahami som dospel k názoru, že rastre by sa dali objektovo reprezentovať pomocou dvojrozmerného poľa. Pri tomto spôsobe riešenia som vychádzal z toho, že väčšina rastrových máp, ktoré sa používajú majú pravidelnú štvorcovú mriežku. Túto pravidelnú mriežku je možné reprezentovať maticou. Tak ako sme uviedli v kapitole 3.1.3, existujú rôzne typy rastrových buniek. Nepravidelné bunky by sa pomocou tohto modelu nepodarilo namodelovať, avšak ako prototyp návrhu objektovo-orientovaného rastru považujem toto riešenie za dostatočné. Pokryjú sa tým všetky základné operácie, ktoré je možné s rastrami vykonávať. Spomenieme v prvom rade reklasifikáciu. Reklasifikácia znamená výber určitých buniek z rastrovej mapy, ktoré spĺňajú zadanú podmienku. Vykonanie takejto operácie je v súlade s maticovými operáciami. Takisto aj operácie prekrývanie rastrových máp. Prekrytie viacerých rastrových máp sa týka výberu rastrových buniek, ktoré spĺňajú nadefinované podmienky. Výsledkom je jeden raster, ktorý zobrazuje len bunky, ktoré vo všetkých prekrývaných rastroch bola splnili podmienky. Medzi tieto podmienky patria tieto typy matematických operácií:

1. aritmetické – sčítanie, odčítanie, násobenie...
2. relačné – porovnávanie hodnôt (menšie, väčšie, rovné...)
3. booleovské – and, or, xor...

Tieto operácie je takisto možné vykonávať maticovými operáciami. Takto nadefinovaná trieda rastrov by obsahovala atribúty podobné ako v prvom riešení, a to rasterName a rasterLegend. Navyše by bol definovaný atribút rasterCells, ktorý by bol typu dvojrozmerného poľa, kde každá položka poľa by obsahovala atribút. Metódy pre prácu s rastrami:

- getRasterName, setRasterName – pre nastavenie, resp. získanie názvu rastra,
- getRasterLegend, setRasterLegend – pre nastavenie, resp. získanie legendy rastra,
- getRasterCells, setRasterCells – pre nastavenie, resp. získanie buniek rastru.

4 Implementácia navrhnutého riešenia

Jedným z cieľov tejto práce je zvolenie vhodného implementačného prostredia, v ktorom demonštrujem mnou definovanú objektovo orientovanú databázu. Pri hľadaní vhodného programovacieho jazyka som vychádzal najmä z jeho podpory pre objektovo orientované programovanie. Ako vhodný programovací jazyk sa ukázal Smalltalk, objektovo orientovaný jazyk. Podstatou smalltalku je jeho objektová povaha. Aplikáciu tvorí množina objektov, ktoré medzi sebou komunikujú zasielaním správ. Objekt po prijatí správy vykoná metódu s rovnomenným názvom. V aplikácií dokonca nemusí byť ani hlavný program. Všetky objektové premenné sú referenciami na ich hodnoty vo virtuálnej pamäti. Používateľ tak nemusí používať pointre do operačnej pamäte. Smalltalk obsahuje aj tzv. garbage collection, mazanie nepotrebných objektov a uvoľňovanie pamäte. Ďalšou výhodou smalltalku je možnosť experimentovania. Je možné za behu upravovať kód, pridávať nové metódy, ktoré sa okamžite preložia a sú používané programom. Pri programovaní je dôležité uvedomiť si, že v smalltalku je všetko objekt. To znamená, že triedy sú rovnocenné objekty, môžu za behu vznikať, meniť sa ale aj mazať. Túto netradičnú vlastnosť umožňujú tzv. metatriedy, špeciálne objekty. Smalltalk je programovací jazyk, ktorý prekladá programy do tzv. byte kódu a ten je za behu prekladaný do strojového kódu. Podrobnejšie opísanie celej charakteristiky smalltalku nie je predmetom tejto práce, preto na oboznámenie čitateľa postačí tento krátky úvod. Viac informácií o smalltalku je možné nájsť na internete [21].

Gemstone je vysoko výkonný používateľský server porovnateľný aj s Oracle databázou. Rádovo umožňuje uložiť miliardy objektov. Základným jazykom tohto systému je Smalltalk DB. Gemstone predstavuje modernú databázu, ktorá umožňuje ukladať aj metódy objektov. To znamená, že sa nejedná len o klasickú databázu ako úložisko dát. Celá aplikácia je rozdelená medzi databázu a VisualWorks. Gemstone spĺňa hneď niekoľko požiadaviek uvedených v kapitole o požiadavkách na databázový systém. Umožňuje vytvorenie distribuovanej databázy, prepojenie niekoľkých databázových systémov, automatické zálohovanie a takisto aj zmenu databázovej schémy.

Do úvahy pri výbere vývojového prostredia pripadali dve možnosti. V oboch prípadoch išlo o použitie jazyka smalltalk. Tým pádom aj princíp obidvoch vývojových prostredí je veľmi podobný, líšia sa prevažne používateľským prostredím a spôsobom pripojenia na databázu Gemstone. Prvá možnosť, ktorou som sa chcel primárne zaoberať, bol Squeak. Squeak je programovací jazyk, ktorý je implementáciou Smalltalku. Výhodou Squeaku je jeho voľná dostupnosť na internete. Takisto medzi výhody patrí aj podpora pre databázu Gemstone. Ovládanie Squeaku je pomerne intuitívne a pôsobí pre používateľov viac user-friendly ako ostatné implementácie Smalltalku. Existuje pomerne veľa priaznivcov Squeaku a teda aj podpora v prípade problémov spojených s implementáciou je na

internete dostatočná. Prpojenie Squeaku na databázu Gemstone je možné po nainštalovaní klientských súborov tejto databázy. Bohužiaľ práve toto bol problém, ktorý zapríčinil, že som namiesto Squeaku, začal uvažovať nad inou variantou. Skúsil som aj nástroj GemTools[1], ktorý je takisto dostupný na internete. Ide v podstate o inštaláciu Squeaku spolu so všetkými knižnicami potrebnými pre pripojenie sa na databázu GemStone. Nanešťastie ani toto nebolo riešenie problému. Pripojenie na databázu vždy skončilo s chybou odvolávajúc sa na službu netldi. Jej úlohou je prijímať požiadavky od GCI klientov (GemTools) a spúšťať tzv. gem proces. Po jeho vytvorení sa gem proces odkazuje naspäť na netldi službu, ktorá mu pridelí GCI klienta žiadajúceho o jeho vytvorenie. Komunikácia prebieha na základe špecifikovaných portov. Na záver netldi služba ukončí spojenie s GCI klientom a tým je vytvorené spojenie GCI klienta s databázou Gemstone. Zo spomínaného princípu fungovania tejto služby je zrejmé, že išlo o pomerne zásadnú chybu. Riešenie, ako odstrániť tento problém, sa mi ani po dlhšej dobe nepodarilo nájsť. Pre overenie, či gemstone server pracuje správne, som vyskúšal iného GCI klienta, konkrétne topaz. Ten je súčasťou každej inštalácie Gemstone. Tento klient nemal žiaden problém s pripojením na vytvorený gem server. Neobjavovala sa spomínaná chyba s netldi službou. Týmto pokusom som si potvrdil hypotézu, že problém bude najskôr v GemTools. Hľadanie chyby by zabralo príliš veľa času a bolo by neefektívne, preto som prešiel na variantu číslo dva. Táto varianta zahrňovala použitie VisualWorks[2]. VisualWorks je integrované vývojové prostredie vytvorené spoločnosťou Cincom pre programovací jazyk Smalltalk. Tento produkt je takisto voľné dostupný na internete v nekomerčnej verzii. Na to, aby bolo možné používať VisualWorks spolu s databázou Gemstone, bolo nutné doinštalovať do VisualWorks GemBuilder. Gembuilder je klient pomocou ktorého Smalltalk komunikuje s databázou Gemstone. GemBuilder poskytuje nástroje pre prácu s databázou podobné tým, ktoré sú poskytované v základnom image-i VisualWorks. Hneď po nainštalovaní GemBilderu do VisualWorks sa podarilo pripojiť na databázu Gemstone. Keďže sa v tejto konfigurácii nevyskytla žiadna zásadná chyba, rozhodol som sa pre implementáciu práve v tomto vývojovom prostredí. Je nutné podotknúť, že som s VisualWorks mal doteraz len minimálne skúsenosti, preto mi bola veľmi nápomocná rozsiahla nápoveda poskytovaná priamo spoločnosťou Cincom.

4.1 Načítanie dát zo shapefile súbora

Načítavanie vstupných dát je realizované načítaním shapefile súboru. Túto možnosť si môže používateľ zvoliť v štandardnom menu cez položku File – Open. Používateľovi sa zobrazí okno pre výber súborov s obmedzením na shapefile súbory, teda súbory s príponou .shp. Výberom súboru sa zavolá metóda readShape s parametrom úplnej cesty k súboru. V prvej časti tejto metódy sa načítava

obsah súboru shapefile. Načítanie binárneho obsahu shapefile súboru je úlohou metódy readBinaryContent. Tá na základe štruktúry shapefile súboru[22] je definovaný if-else príkaz, ktorý rozhoduje, objekt akého typu sa vytvorí. V praxi sa stretneme najčastejšie s týmito typmi a im pridelenými hodnotami v shapefile štruktúre:

- 0 – žiadny geoobjekt,
- 1 – bod (point),
- 3 – reťazec línií (polyline),
- 5 – polygón (polygon).

Pochopiteľne zo shapefile súboru je možné načítať aj objekty iných zložitejších typov. Nám pre ukážku postačia objekty, pre ktoré máme nadefinované triedy v objektovo-orientovanej databáze. Po načítaní hodnoty daného typu objektu sa jednoducho zavolá metóda, ktorá príslušný objekt vytvorí v databáze zo všetkými jeho vlastnosťami. Druhá časť metódy obsahuje kontrolu existencie súboru .shx, ktorý obsahuje informácie o priestorových tvaroch zo shapefile súboru. Na základe zistenej cesty k súboru sa odsekne prípona .shp a nahradí príponou .shx s následným overením, či naozaj tento súbor exituje. Ak neexistuje, tak sa používateľovi zobrazí chyba. V opačnom prípade sa všetky objekty zo shapefile súboru úspešne uložia do databázy Gemstone.

4.2 Triedy v databáze Gemstone

V objektovo-orientovanej databáze Gemstone sú definované základné triedy definované v kapitolách 5.1 a 5.2. Aby sme predišli problémom pri ukladaní objektov do databáze, definoval som obmedzenia pre jednotlivé atribúty geoobjektov nasledovne.
Trieda bod (Point):

- coordX: float – súradnica X v súradnicovom systéme,
- coordY: float – súradnica Y v súradnicovom systéme,
- pointType: String – typ bodu (uzol, koncový bod...),
- startNode: boolean – true/false,
- endNode: boolean – true/false,
- pointBelongsTo: bez obmedzenia, môže nadobúdať referencie na rôzne objekty.

Trieda línia (Line):

- startPoint: Point – začiatočný bod,

- endPoint: Point – koncový bod,
- leftNeighbour: Polygon – ľavý sused, referencia na objekt typu polygón,
- rightNeighbour: Polygon – pravý sused, referencia na objekt typu polygón,
- lineBelongsTo: bez obmedzenia – línia môže patriť objektom rôznych typov,
- lineLength: float – dĺžka línie,
- crossObject: bez obmedzenia – líniu môžu pretínať objekty rôznych typov.

Trieda reťazec línií (PolyLine):

- lineNumber: int – počet línií tvoriacich reťazec línií,
- lines: Line – referencie na všetky objekty typu línia tvoriace reťazec línií,
- crossObject: bez obmedzenia – reťazec línií môžu pretínať objekty rôznych typov.

Trieda polygón (Polygon):

- lines: Line – referencie na línie tvoriace objekt typu polygón,
- startEndNode: Point – referencia na objekt typu bod,
- perimeter: float – obsah,
- centroid: Point – referencia na objekt typu bod ,
- crossObject: bez obmedzenia - lpolygón môžu pretínať objekty rôznych typov.

Trieda raster (Raster):

- name: String – názov rastru,
- rasterLegend: Array – legenda reprezentovaná dvojrozmerným poľom,
- rasterCell: Array – bunky rastra reprezentované dvojrozmerným poľom.

Presné definície tried v databáze Gemstone sú súčasťou prílohy 1. Pre ľahšiu manipuláciu s týmito triedami som urobil importovací skript vo formáte topaz, ktorý bude priložený k výslednej práci.

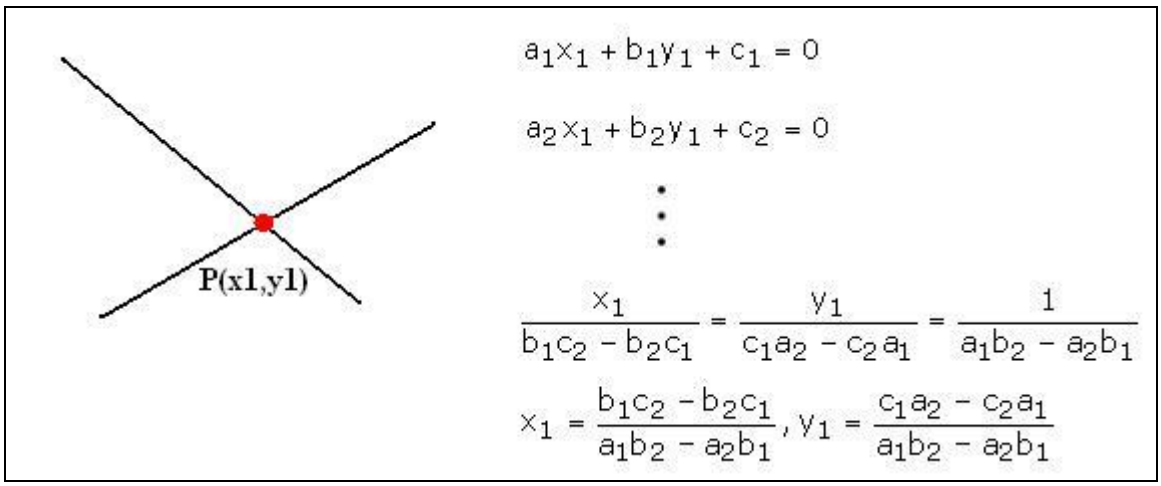
4.3 Aplikácia JK GIS

Prototyp samotnej aplikácie sa skladá z dvoch častí. Prvou je hlavné okno aplikácia, ktoré obsahuje štandardné hlavné menu, tlačidla pre spúšťanie rôznych akcií a textové pole, do ktorého sa vypisujú informácie o aktívnom objekte. Dôležitou funkciou je Compute topology. Ide o funkciu, ktorá vypočítavá priesečníky geoobjektov – línií a ukladá ich ako referencie do príslušných objektov. Tak

ako polygón, tak aj reťazec línií obsahuje referencie na línie, z ktorých je zložený. Vypočítaním priesečníkov týchto línií sa určia topologické vzťahy všetkých objektov.

4.3.1 Výpočet priesečníkov línií

Pri výpočte priesečníkov línií som vychádzal z matematických relácii, ktoré platia pre výpočet priesečníka dvoch línií v dvojrozmernom priestore. Matematický výpočet je uvedený na obr. 4.1.



Obr. 4.1: Matematický výpočet priesečníka.

Vstupnými argumentmi algoritmu sú dva objekty typu Line, kde súradnice x0, y0, x1 a y1 sú súradnice bodov jednej línie a súradnice x2, y2, x3 a y3 sú súradnice druhej línie. Algoritmus sa skladá z troch častí. V prvej časti sa vyrátajú premenné a, b, c. V druhej časti sa vypočíta determinant: A nakoniec sa vypočítajú súradnice priesečníka. Zápis je v syntaxi jazyka smalltalk na obr. 4.2.

```
a1 := (y1 - y0) / (x1 - x0).
a2 := (y3 - y2) / (x3 - x2).
b1 := -1. b2 := -1.
c1 := (y0 - (m1 * x0)).
c2 := (y2 - (m2 * x2)).

det := 1 / ((a1 * b2) - (a2 * b1)).

p1 := (((b1 * c2) - (b2 * c1)) * det).
p2 := (((a2 * c1) - (a1 * c2)) * det).
```

Obr. 4.2: Algoritmus výpočtu priesečníka dvoch línií.

Pre objekty typu polygón a reťazec úsečiek nachádzajúcich sa v databáze je možné napísať nasledujúci pseudokód uvedený na obr. 4.3.

```
Pre všetky objekty v databáze
    Je objekt typu polyline
        Pre všetky polyline.lines x
            Vypocitaj_priesecnik(x, ostatné línie)
            Existuje priesečník
                Ulož referenciu na líniu
    Je objekt typu polygon
        Pre všetky polyline.lines x
            Vypocitaj_priesecnik(x, ostatné línie)
            Existuje priesečník
                Ulož referenciu na líniu
```

Obr. 4.3: Pseudokód určovania topológie geoobjektov.

4.3.2 Dotazy na geoobjekty

Po určení topológie je možné pracovať s geoobjektami a vykonávať rôzne dotazovacie úlohy zamerané na ich atribúty, resp. topológiu. Dotaz typu nájdi všetky rieky (línie), ktoré sa vlievajú do určitej rieky (línie) obsahuje funkcia getCrossedRivers uvedená na obr. 4.4.

```
getCrossedRivers: aRiver
| crossedRivers |
crossedRivers := Rivers select:[ :tRiver | (tRiver crossObjects) = aRiver ].
^crossedRivers
```

Obr. 4.4: Príklad dotazu na pretínajúce objekty.

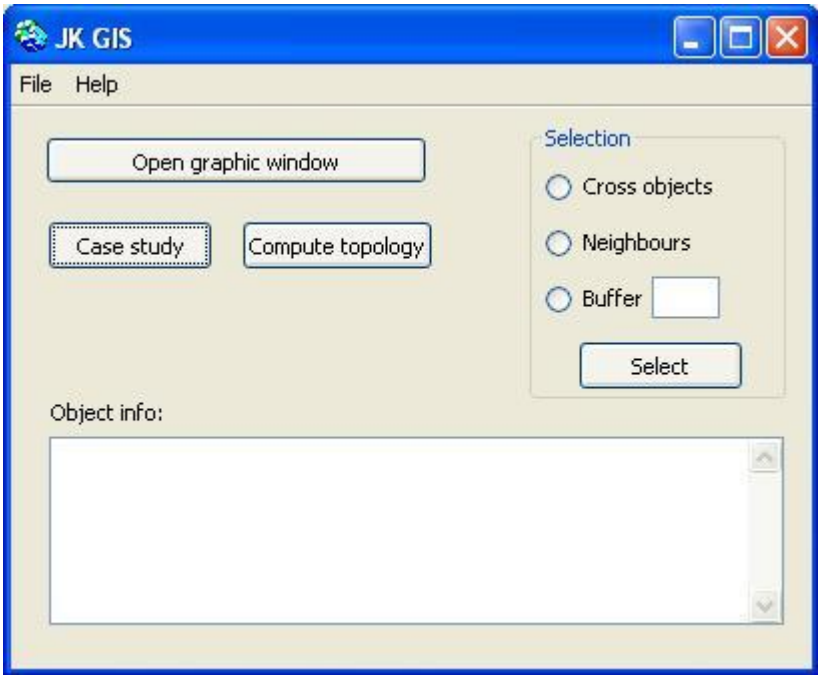
Príkladom dotazu na geoobjekt typu polygón dotazujúci sa na konkrétny atribút môže byť nasledovné. Zobraz všetky polygóny (pozemky) s obvodom (perimeter) väčším ako X m^2. (obr. 4.5)

```
getPolygonWithPerimeter: aPerimeter
| selectedPolygons |
selectedPolygons := Polygons select:[ :tPolygon | (tPolygon perimeter) >= aPerimeter ].
^selectedPolygons
```

Obr. 4.5: Príklad dotazu na atribút geoobjektu.

4.3.3 Vzhľad aplikácie

Okno aplikácie je zobrazené na obr. 4.6. a pozostáva zo základného lištového menu, v ktorom je možné aplikáciu ukončiť, načítať dáta, prípadne zobraziť nápovedu. Grafické výsledky sa zobrazujú v okne, ktoré sa otvára tlačidlom *Open graphic window.* K dispozícii sú možnosti geografických analýz, a to získanie geoobjektov pretínajúcich označený geoobjekt, získanie susedných geoobjektov a vytvorenie bufferu okolo geoobjektu. Výsledky o označenom geoobjekte sa zobrazujú v textovom poli. V prípade načítania nových geoobjektov existuje možnosť dopočítať topologické vzťahy cez tlačidlo *Compute topology.* Na záver je pripravená prípadová štúdia, ktorá sa zobrazí po kliknutí na tlačidlo *Case study*.



Obr. 4.6: Okno aplikácie JK GIS.

5 Prípadová štúdia – GIS ako nástroj pre podporu rozhodovania

Takmer v každej oblasti každodenného života, či už. v IT, hospodárstve, ekonomike, vede a mnohých ďalších, si človek pri riešení úloh vyberá z viacerých možností vedúcich k optimálnemu riešeniu. Aby takéto rozhodovanie nebolo založené len na skúsenostiach odborníkov, existuje množstvo nástrojov pre podporu rozhodovania. Jedným z nich sú aj GIS systémy. Každý nástroj pre podporu rozhodovania, a teda aj GIS systém, sa používa v prípade, ak:

- existuje viacero alternatív správania sa účastníka,
- alternatívy nie sú navzájom rovnocenné,
- existuje jedno alebo viac kritérií efektívnosti, podľa ktorých je možné posúdiť, ktorá alternatíva je vhodnejšia.

V prípade použitia GIS systémov ako nástrojov pre podporu rozhodovania môžeme hovoriť o tzv. systéme pre podporu priestorového rozhodovania. Zjednodušene povedané ide o spojenie GIS systémov a systémov pre podporu rozhodovania, tzv. DSS systémov. Jedna z definícií DSS systémov hovorí, že DSS je ľubovoľný interaktívny systém, ktorý zobrazuje a spracováva dáta a tým napomáha pri riešení rozhodovacích úloh [17]. Medzi jeho základné vlastností patria:

- používateľ nepotrebuje špeciálne vedomostí z výpočtovej techniky na jeho obsluhu, je teda user-friendly,
- zobrazuje výsledné informácie vo forme blízkej používateľovi,
- nezaťažuje používateľa zbytočnými informáciami, používateľ nemusí poznať všetky jeho funkcie, iba tie, ktoré potrebuje.

Prínosom spojenia GIS a DSS systémov je geografické hľadisko pri podpore rozhodovania. Geografické informácie majú svoju nespornú hodnotu. Všeobecne akceptovaný odhad totiž hovorí, že až 60 - 80 % ľudských rozhodnutí je ovplyvnených priestorom, resp. takýto podiel informácii má priamy alebo sprostredkovaný priestorový kontext. Informácia má veľmi rozdielnu hodnotu pre rôznych používateľov. GIS systém ako nástroj pre podporu rozhodovania našiel uplatnenie v mnohých oblastiach.

Stručný opis problematiky

V existujúcich riešeniach systémov pre podporu rozhodovania, v ktorých sa uplatňujú GIS systémy, sa využívajú vektory a rastre ako základné prvky systému. Tento trend je ovplyvnený samotným vývojom GIS systémov. Niektoré GIS systémy, ktoré slúžia pre podporu rozhodovania, využívajú dokonca iba vektorovú reprezentáciu. Tá síce prináša pre niektoré projekty dostatočnú mieru priestorovej informácie, ale nemá takú rozhodovaciu silu ako spojenie vektorovej a rastrovej reprezentácie. Cieľom tejto prípadovej štúdie, ako aj celej diplomovej práce, je ukázať výhody objektovo-orientovanej reprezentácie GIS systému. Keďže každý proces rozhodovania so sebou nesie určité riziká a jeho výsledok je spojený s čerpaním zdrojov (finančných, prírodných atď.), zníženie týchto zdrojov zavedením objektovo-orientovanej metodiky do GIS systémov by bol prínosom. Uvediem niekoľko výhod, ktoré by GIS systém slúžiaci pre podporu rozhodovania mohol získať.

1. Reálnejšie chápanie priestoru z pohľadu bežného používateľa.

Rozhodovanie je zložitý proces, pri ktorom je nutné brať v úvahu množstvo faktorov ovplyvňujúcich samotné rozhodnutie. V našom prípade k tomu musíme začleniť aj priestorové faktory. Chápanie modelu sveta vo forme vektorov a rastrov nemusí byť pre každého triviálne. Preto je vhodné uvažovať o jednotlivých aspektoch sveta ako o objektoch. Ideálny stav pre používateľov je reprezentovať reálne objekty sveta tak, ako ich oni sami v skutočnosti vidia, alebo sa aspoň čím viac k tomu priblížiť. Tým by odpadala akákoľvek nutnosť technických vedomostí o GIS systémoch. Rozhodovanie na základe priestorových informácií by sa tak mohlo zjednodušiť.

2. Zjednodušenie GIS systému ako nástroja pre podporu rozhodovania.

Nespornou výhodou, ktorá bola spomínaná v predchádzajúcich kapitolách, je samotná objektovo-orientovaná implementácia GIS systému. Nahradenie relačnej databázy z množstvom tabuliek za objektovo-orientovanú. Údaje o každom objekte sú zapuzdrené v rámci objektu a nie sú zahrnuté v niekoľkých relačných tabuľkách, ktoré pri veľkom množstve spôsobujú pomerne veľkú neprehľadnosť.

3. Uplatnenie moderného trendu objektovo-orientovanej reprezentácie.

Samotná objektovo-orientovaná reprezentácia zatiaľ nenašla v GIS systémoch veľké uplatnenie. Súčasný trend však zvýrazňuje jej výhody, a preto aj jej zavedenie do GIS systémov by mohlo byť prínosom. Podobne aj využívanie objektovo-orientovaných databáz, ktoré by uľahčili najmä vytváranie nových geodatabáz. Nová architektúra, nová metodika a celkovo reálnejšie ponímanie a zachytávanie geobjektov by malo poslúžiť k jednoduchšej aplikácii GIS systémov praxi. Pri samotnom využití GIS systémov pre podporu rozhodovania by zavedenie týchto nových metodík

malo za následok zjednodušenie práce pre samotných používateľov ako aj vývojárov týchto systémov.

Súčasný stav

V súčasnosti sa využívajú GIS systémy ako nástroj pre podporu rozhodovania v mnohých odvetviach každodenného života. Preto si zanalyzujeme ich súčasný stav, ich funkcie a prínos. V prvom rade práca s GIS by mala zjednocovať tri aspekty – kartografický, databázový a analytický. Podľa povahy aplikácie však môže niektorý z nich výrazne dominovať.

Jednou z oblasti kde sú GIS systémy veľmi často využívané, je pôdohospodárstvo. Produkt s názvom IGIS RP [18] našiel uplatnenie ako integrovaný informačný systém rezortu pôdohospodárstva SR. Primárnu funkciu, ktorú plní je funkcia umožňujúca podporu rozhodovania v procese optimalizácie využívania krajiny z hľadiska poľnohospodárskeho, lesohospodárskeho a vodohospodárskeho. Jeho údajový model predstavuje abstrakciu objektívnej reality. Je základom pre budovanie relačnej databázy, s ktorou sa stretávame vo väčšine súčasných GIS systémov. Model obsahuje iba tie prvky reálnej krajiny a ich vlastnosti, ktoré sú adekvátne z hľadiska potrieb. Tým splňuje jednu z vlastností systémov pre podporu rozhodovania – poskytovať používateľovi len informácie, ktoré sú pre daný proces rozhodovania dôležité Predmetom záujmu tohto modelu sú fenomény priamo alebo nepriamo spojené s riadením pôdohospodárstva, patriace k vodám, pôde (vlastnosti pôdy a využitie pôdy), lesom, výškopisu. Hlavné zložky dopĺňajú sekundárne, umelé prvky krajiny - územnosprávne celky a informácie katastra nehnuteľností. Základnými elementmi údajového modelu sú:

- ***entita*** **-** fenomén objektívnej reality, ktorý je ďalej nedeliteľný,
- ***atribút*** **-** vlastnosť entity,
- ***doména*** **-** definuje dátový typ atribútu entity a vyjadruje množinu (obor hodnôt) všetkých prípustných hodnôt, ktoré môže daný atribút nadobúdať,
- ***väzba*** **-** popisuje vzájomný vzťah entít.

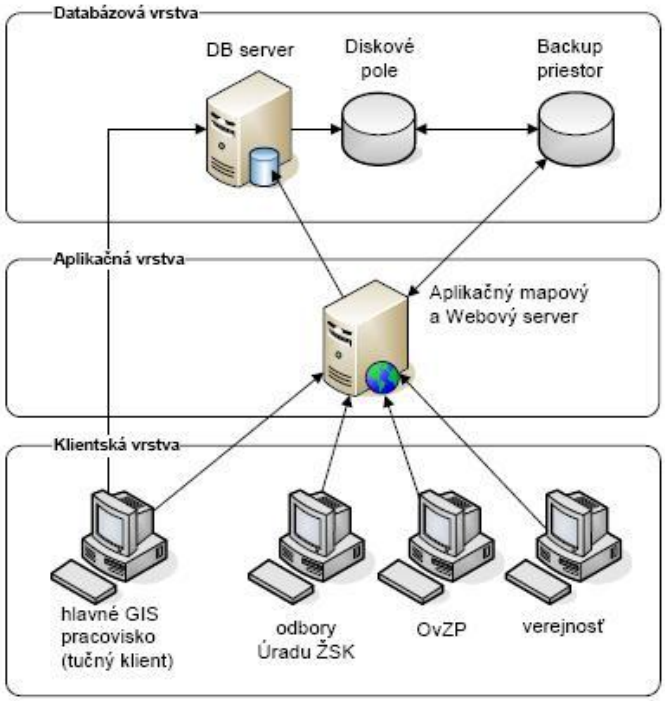
Spomínaný GIS systém využíva pre zobrazovanie geobjektov vektory a rastre. Vo väčšej miere sú informácie čerpaná z rastrov, ktoré sú využité na priestorové analýzy. Typicky sa jedná o analýzu určitého územia (priestoru), kde sa zohľadňujú viaceré rastrové vrstvy. Keďže sa jedná o GIS systém, s ktorým môžu pracovať naraz viacerí používatelia cez internet, musel sa vyriešiť problém zdieľania dát z rôznych zdrojov. Riešením bola platforma OpenGIS (Open Geodata Interoperability Specification), ktorá je produktom, resp. bola sformulovaná OGC (Open GIS Consortium) a rieši tento problém z viacerých hľadísk – technického, informačného a organizačného. Platforma OpenGIS

je podľa OGC definovaná ako otvorené a vzájomne spolupracujúce spracovanie geografických informácii, alebo schopnosť zdieľať prehľadne heterogénne dáta a geoinformačné zdroje v sieťovom prostredí. Podľa predpokladov tento GIS systém pre podporu rozhodovania pracuje na očakávanom princípe – reprezentácia geoobjektov pomocou rastrov a vektorov spolu s ich ukladaním do relačnej databázy.

Ďalším GIS systémom, ktorý plní aj funkciu rozhodovania, je geografický informačný systém Žilinského samosprávneho kraja (GIS ŽSK). Tento systém predstavuje databázový a mapový informačný systém na tvorbu, uchovávanie, analýzu a poskytovanie geopriestorových informácií. GIS ŽSK predstavuje sofistikovaný nástroj na podporu rozhodovania zamestnancov Úradu ŽSK ako aj na prezentáciu digitálnych informácií z regiónu ŽSK širokej laickej, ale i odbornej verejnosti [19]. Systém ma implementované nasledujúce funkcie:

- funkcie na podporu rozhodovania,
- funkcie na vyhľadávanie (jednoduché, hierarchické, tvorba dopytov),
- funkcie pre prácu s mapou (rôzne spôsoby grafického výberu, lokalizácia objektov, vyhľadávacie zóny a pod.),
- on-line editovanie objektov (grafická a atribútová časť),
- tvorba náčrtkov (komunikácia a zdieľanie informácií o priestorových javoch/objektoch medzi užívateľmi).

Systém je postavený na trojvrstvovej architektúre, ktorá sa skladá z databázovej, aplikačnej a klientskej vrstvy ako je možné vidieť na obr. 5.1. Databázový systém využívaný v ŽSR GIS je distribuovaná relačná databáza Oracle. Platforma, ktorá je používaná v systéme, je základná platforma ESRI - ArcGIS, ArcSDE, ArcIMS.



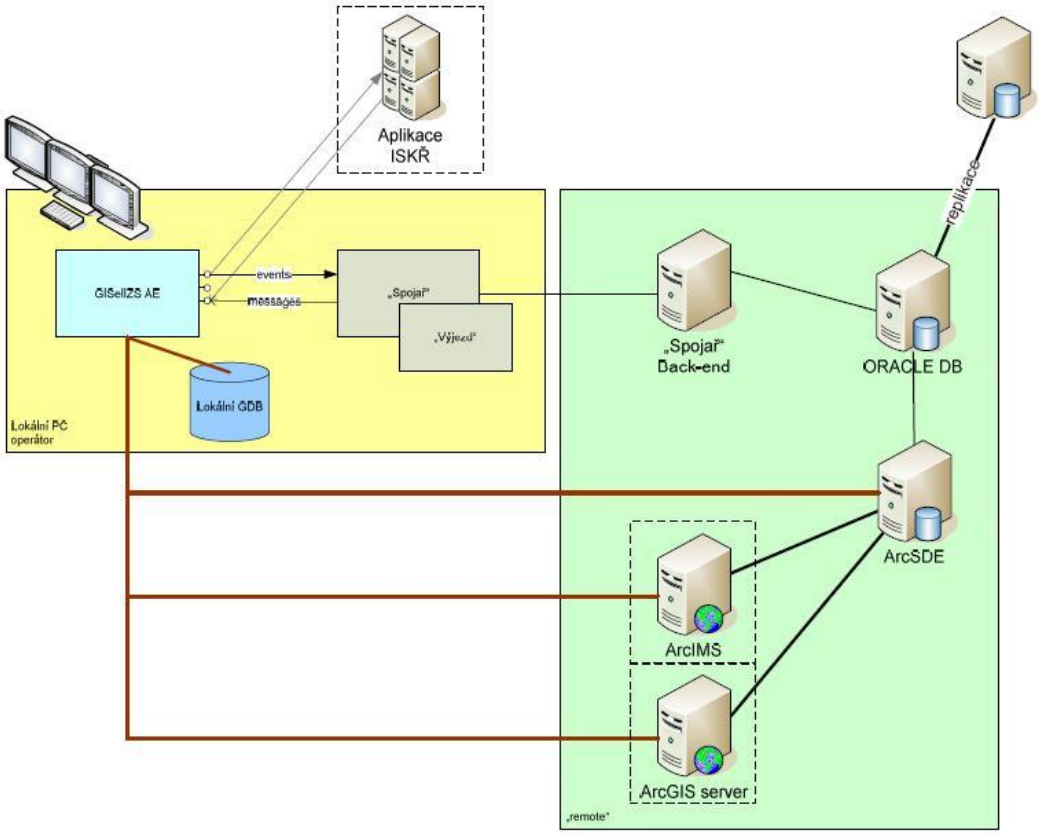
Obr. 5.1: Architektúra GIS systému ŽSR GIS [19].

Posledný príklad, ktorý si uvedieme, opäť demonštruje súčasný stav využívania GIS systémov. Je to aplikácia GISellZS AE [20]. Táto aplikácia je využívaná v hasičskom záchrannom zbore (HZZ). Potenciál využitia GIS systému práve v HZZ je obrovský. Pri každom zásahu HZZ je nutné najprv správne lokalizovať miesto výjazdu a veľmi dobre poznať terén. GIS v tomto prípade zohráva aj úlohu podpory pre rozhodovanie. Podľa miest výjazdu je možné vopred určiť prostriedky, ktoré budú pri zásahu nutné a pod. Dôležité je poznať dostupnosť terénu, t.j. jeho sklon, typ pôdy, cestnú, riečnu ale takisto aj kanalizačnú sieť. Architektúra celého systému je postavená na platforme GIS ISKŘ, ktorá splňuje nasledujúce kritéria:

- platforma ESRI,
- využitie databázového systému ORACLE v spojení s middleware ArcSDE,
- využitie replikačných databázových mechanizmov,
- využitie internetových aj desktopových technológií.

Základom architektúry na obr. 5. 2 je dátový sklad obsahujúci údaje z celého územia. Tento sklad je replikovaný do viacerých pracovísk obsahujúcich kópiu dátového skladu. Je to z dôvodu nezávislého používania tohto systému viacerými používateľmi z viacerých miest naraz. Aplikácia GISellZS AE je vytvorená na platforme Microsoft Windows .NET. Využíva mapové programové komponenty ESRI

ArcObjects. Aplikácia podporuje systémy riadenia bázy dát pre vyhľadávanie objektov sú Oracle 9i a 10g, SQL Server 2000 a SQL Server 2005.



Obr. 5.2: Nasadenie aplikácie GISellZS AE v rámci architektúry GIS ISKŘ [20].

Takto navrhnutá aplikácia spĺňa základné funkcie GIS systému:

- poskytuje informácie o území,
- vyhľadáva objekty v priestore,
- umožňuje vyhľadávanie dát z relačnej databázy,
- poskytuje prehľadné používateľské rozhranie s intuitívnym ovládaním.

Zhrnutím uvedených súčasných GIS systémov môžeme odvodiť nasledujúce závery. Z uvedených GIS systémov pre podporu rozhodovania vyplývajú určité spoločné črty:

- všetky využívajú vektory a rastre na reprezentáciu geoobjektov,
- medzi databázami slúžiacimi na uchovávanie geoobjektov dominujú relačné databázové systémy,
- sú implementované základné priestorové funkcie a analýzy,

- systémy obsahujú informácie len o geoobjektoch, ktoré priamo súvisia s využívaním systému,
- umožňujú zdieľanie dát z rôznych zdrojov,
- väčšina systémov ma webové rozhranie s intuitívnym ovládaním.

Zo spoločných čŕt skúmaných GIS systémov je zrejmé, že objektovo-orientovaná koncepcia sa zatiaľ v súčasných GIS systémoch nepresadila. Dôraz je kladený na zaužívané štandardy v tejto oblasti. Vo všetkých skúmaných systémoch sa dodržujú základné fázy tvorby GIS systémov. Od zberu údajov, spracovanie údajov, priestorových analýz až po zobrazovanie výsledkov a štatistík. Ďalšie spracovanie údajov a ich ukladanie do DB, či už relačnej alebo objektovo-orientovanej, závisí od toho koľko výhod, resp. nevýhod daný spôsob spracovania dát poskytuje. Na základe analýzy súčasného stavu uvediem výhody aj nevýhody relačného prístupu v GIS systémoch.
Výhody:

- lacnejšie databázové relačné systémy,
- definované štandardy v oblasti GIS,
- postačujúca miera informácií získaná z analýz údajov,
- spôsob využitia dát nezávisí od spôsobu uloženia,
- explicitné definovanie vzťahov medzi lokalizačnými a atribútovými údajmi,
- k dispozícii sú automatizované nástroje na manipuláciu s dátami.

Nevýhody:

- potreba súčasnej práce s veľkým počtom súborov,
- slabšia miera abstrakcie geoobjektov reálneho sveta,
- komplikovanejšia manipulácia,
- zložitá reprezentácia niektorých priestorových vzťahov,
- zložitá tvorba hierarchií objektov,
- logické modelovanie sa vzďaľuje od konceptuálneho (údaje jedného objektu sú vo viacerých relačných tabuľkách).

Aplikovanie objektovo-orientovaného princípu na GIS systém pre podporu rozhodovania
Objektovo – orientovaný princíp aplikujeme na GIS systém, ktorý bude slúžiť potenciálnym investorom pri rozhodovaní, ktorá časť krajiny je najvhodnejšia pre ich podnikanie. Do úvahy budeme brať len geoobjekty, ktoré priamo súvisia s takýmto typom rozhodovania. Základné geoobjekty budú reprezentované pomocou objektovo – orientovaných vektorov a rastrov.

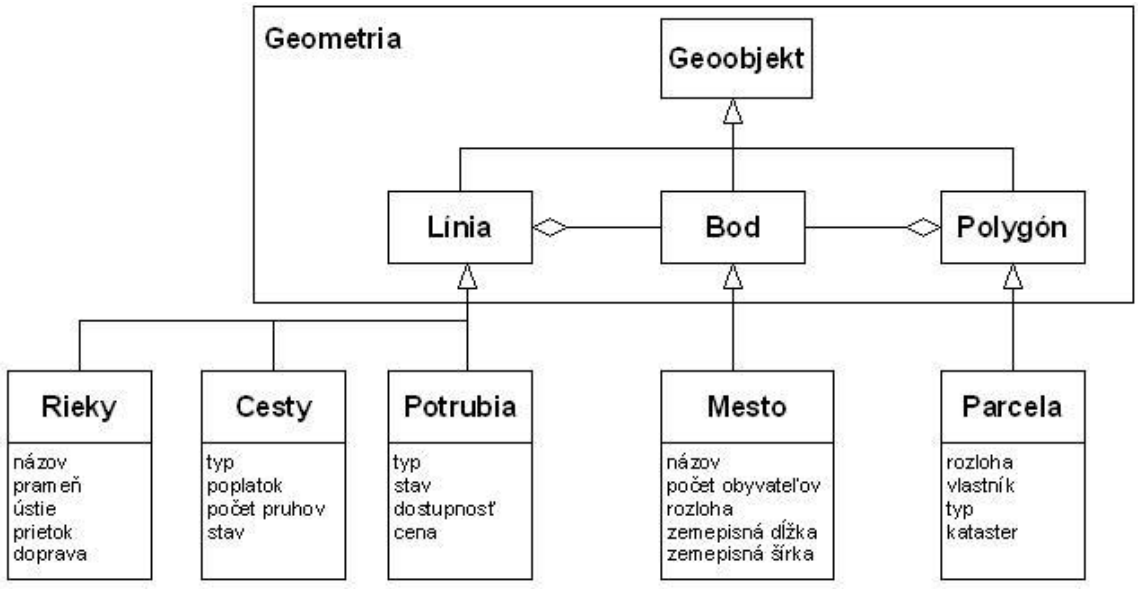
Medzi vektory zaradíme:

- cestnú sieť (cesty a diaľnice),
- riečnu sieť,
- inžinierske siete,
- hranice pozemkov, parciel,
- mestá, obce .

Medzi rastre zaradíme:

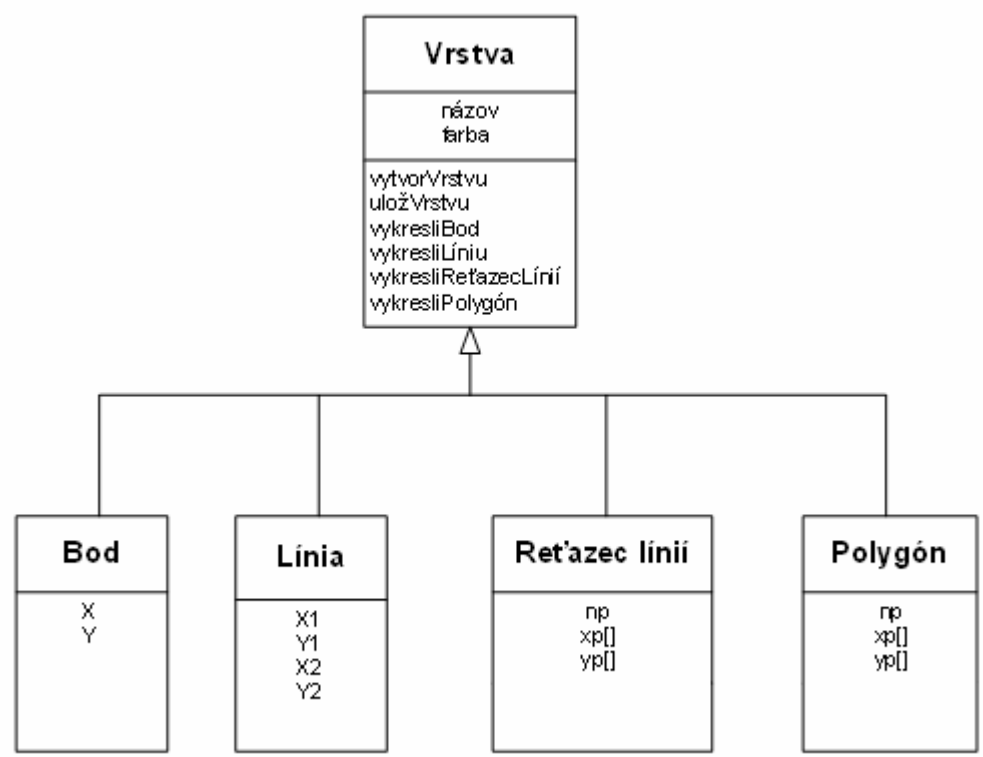
- hustotu obyvateľstva,
- mieru nezamestnanosti,
- typ terénu,
- klimatické podmienky.

Každý vektor spomenutý vyššie bude dediť geometrické vlastnosti podľa toho aký typ predstavuje a tak isto bude mať aj svoje jedinečné vlastnosti. Schéma vektorov s ich vlastnosťami je znázornená na obr. 5.3.



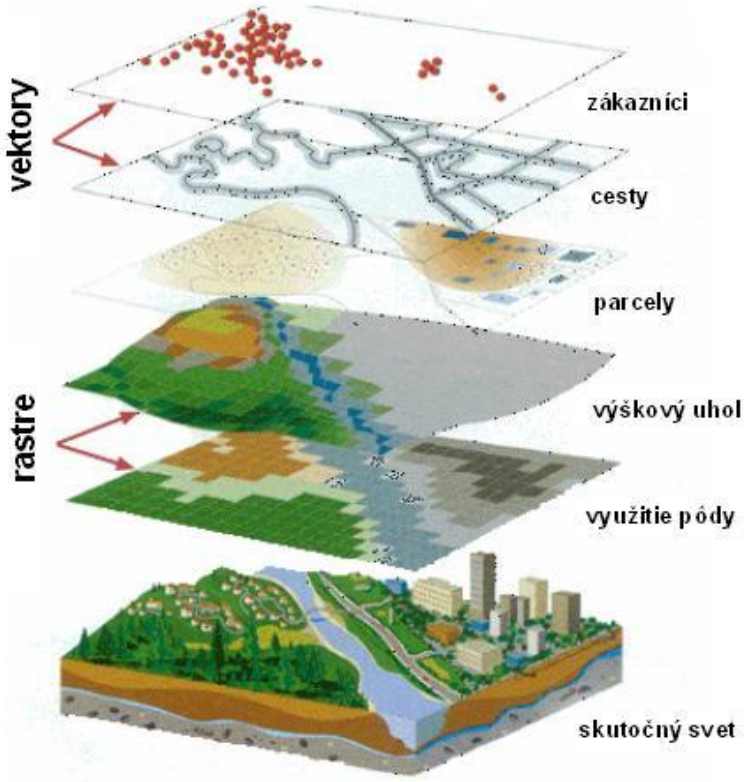
Obr. 5.3: Schéma vektorových geoobjektov a ich vlastností.

Rovnaké typy geoobjektov sú zoskupované do vrstiev. Každá vrstva môže znázorňovať buď body, línie, reťazce línií alebo polygóny znázornené na obr. 5.4. Jednotlivé vrstvy sa môžu medzi sebou prekrývať a tým sa zlučovať do nových vrstiev.



Obr. 5.4: Diagram reprezentujúci vrstvu a jej objekty.

Potenciálne rozloženie reálneho sveta do jednotlivých vrstiev je uvedené na obr. 2.10. Samozrejme, že samotný reálny svet je oveľa zložitejší, ale pre ukážku nám obr. 2.10. postačí. Počet vrstiev, ktoré reprezentujú reálny svet môže byť rôzny. Stupeň abstrakcie zobrazovania reálneho sveta závisí od požiadaviek na daný GIS systém. Na našom príklade je reálny svet rozdelený do piatich vrstiev. Z toho sú tri vektorové vrstvy a dve rastrové vrstvy. To, ktorú vrstvu si používateľ zobrazí, resp. koľko vrstiev naraz, závisí len od jeho kritérií. Tým, že sa niektoré vrstvy pri zobrazovaní vynechajú má za výsledok prehľadnejšiu analýzu skúmaných dát bez nadbytočných informácií (geoobjektov). Ako je možné vidieť z obr. 5.5. prekrývať sa môžu vektorové a rastrové vrstvy navzájom. Táto skutočnosť je pravidelne využívaná pri rôznych priestorových analýzach. Spravidla riešime úlohy, ktoré sa netýkajú len samotných vektorov ale najmä javov (rastrov), ktoré sa na danom území vyskytujú. Preto je možné zodpovedať napr. dotazy typu, ktoré mestá ležia na povrchu so sklonom do 10 stupňov. Ako raster sa použije vrstva výškový uhol (angl. elevation) a vrstva reprezentujúca vektory teda mestá.



Obr. 5.5: Rozdelenie reálneho sveta do vrstiev.

Objektovo orientovaný GIS systém je schopný odpovedať na priestorové dotazy, ktoré potenciálny investor potrebuje. GIS systém dokáže:

- určiť zemepisné súradnice ľubovoľného bodu,
- zistiť vzdialenosť medzi dvoma ľubovoľnými bodmi,
- vykonávať základné priestorové analýzy,
- prehľadne zobraziť geoobjekty.

Na záver tejto štúdie si uvedieme niektoré výhody objektovo orientovaného prístupu v porovnaní s relačným. Teda prečo by bolo modelovanie objektovo orientovaného GIS systému jednoduchšie:

- väčšia podobnosť a prepojenie konceptuálneho a logického modelu,
- jeden objekt predstavuje jeden súbor,
- jednoduchá tvorba hierarchií a dedičností objektov,
- explicitne definovanie objektov - identita, štruktúra a správanie,
- zjednotenie polohových a atribútových údajov do jednej databázy,

- využitie integrovaného jazyka a komunikácie pomocou správ,
- jednoduché vytváranie väzieb na grafické prostredie,
- využitie samotnej objektovo-orientovanej metodiky.

5.1 Modelový príklad

Ukážeme si modelový príklad, ktorý objektovo-orientovaný GIS systém dokáže spracovať. Potenciálny investor hľadá provinciu v štáte Kanada pre vybudovanie špedičného skladu. Provincia musí spĺňať nasledujúce kritéria:

- provincia má aspoň 1 000 000 obyvateľov,
- provinciou musí prechádzať cesta 1. triedy,
- prístav v okolí 100 km výhodou,
- rovinatý terén,
- preferovaný je juh krajiny.

Modelový príklad potrebuje vektorové dáta pre letiská, cesty a prístavy v štáte Kanada a rastrové dáta pre terén v štáte Kanada (za poplatok) dostupné z [23]. Letiská a prístavy sú vektorové dáta typu bod a cesty sú vektorové dáta typu reťazec línií. Príklad si rozdelíme do 4. krokov.

1. krok

Na základe kritéria – v provincia má aspoň 1 000 000 obyvateľov – si označíme všetky provincie (polygóny), ktorých atribút početObyvateľov je väčší ako 1 000 000 prostredníctvom dotazu:

```
| selectedProvince |
selectedProvince := Provinces select:[ :tPolygon | (tPolygon population) >= 1000000 ].
^selectedPolygons
```

Týmto krokom máme zobrazené všetky provincie s počtom obyvateľov nad 1 000 000.

2. krok

Ďalšie kritérium – provinciou musí prechádzať cesta 1.triedy – splníme nasledujúcim dotazom:

```
| road1class provinceWithRoad |
road1class := Roads select: [ :tRoad | (tRoad type) = 'class1' ].
provinceWithRoad := Provinces select:[ :tPolygon | (tPolygon crossObjects) = Road1class ].
^ provinceWithRoad
```

Najprv boli vybrané všetky cesty, ktoré sú typu 1. triedy. Je to dotaz na konkrétny atribút geoobjektu. Následne sa kolekcia vybraných geoobjektov (ciest 1. triedy) použila v druhom dotaze. Dotaz prehľadal všetky referencie geoobjektov provincii na objekty, ktoré ich pretínajú. Referencie sú v tomto prípade uložené v atribúte crossObjects. V prípade, že danú provinciu pretína niektorá cesta 1. triedy bola táto provincia vybraná.

3. krok

Tretie kritérium – prístav v okolí 100 km – sa realizuje na základe priestorovej operácie, tzv. buffer. Keďže prístavy sú reprezentované bodmi, ich okolie do 100 km sa znázorní ako kružnica s polomerom 100 km, kde stredom kružnice je daný prístav – bod. Dotaz vyzerá nasledovne:

```
|circle provincedWithRoadsAndPort|
Pre všetky Pristavy
    circle := Circle center: aPristav radius: 100.
    provinceWithRoadAndPort := Provinces select:[ :tPolygon | (tPolygon crossObjects) = circle ].
^ provinceWithRoadAndPort
```

Výsledkom sú provincie, ktoré obsahujú cesty 1. triedy a ktorých dostupnosť do prístavu je 100 km.

4.krok

Posledný krok je zameraný na rastrový dotaz. Úlohou je zistiť, v ktorých provinciách sa nachádza nížinatý terén. Z rastru CanadaUseLand sú získané len tie hodnoty 2-rozmerného poľa, ktoré obsahujú atribút nížiny.

Na záver je pre potencionálneho investora zoznam provincií, ktoré spĺňajú všetky jeho požadované kritéria a je na ňom, ktorú z týchto provincií si vyberie.

6 Záver

Diplomová práca detailne popisuje problematiku objektovo-orientovaných GIS systémov. V jej samom úvode je popísaný úvod do geografických informačných systémov, od zberu dát až po ich zobrazovanie. Následne je dôraz kladený na analýzu súčasného stavu geoobjektov do databáz. Pri ukladaní dát je poukázané na zaužívané ukladanie dát do relačných databázových systémov a následný prechod na objektovo-orientovanú databázu. Pred samotným návrhom objektovo-orientovaného GIS systému bolo potrebné naštudovať odbornú literatúru so zameraním na objektovo-orientovanú metodiku. Táto časť patrila medzi najťažšie, keďže literatúry venujúcej sa tejto problematike je veľmi málo. Objektovo-orientovaný GIS je len na začiatku vývoja a v praxi sa zatiaľ takmer nevyskytuje. Vytvoriť si obraz o tejto problematike bolo preto veľmi náročné a výsledkom sú rôzne objektovo-orientované metódy, ktoré sa môžu využiť v GIS a sú popísané v práci. Na konci tejto etapy bola zostavená súvislá analýza objektovo-orientovaného GIS, ktorá môže byť prínosom pre budúce pokračovanie v tejto oblasti.

V ďalšej fáze diplomovej práce boli zanalyzované dostupné objektovo-orientované databázy spolu so zostavením požiadaviek, ktoré by mala spĺňať objektovo-orientovaná databáza určená pre GIS. Po výbere vhodnej databázy prebiehal návrh uloženia geoobjektov do databázy. Pri návrhu ukladania geodát do databázy je vychádzané z metodologickej časti, ktorá popisuje spôsob ukladania objektovo-orientovaných geoobjektov pri zachovaní ich topológie. Poznatky boli čerpané najmä z [6],[9],[10],[11] a [14], ktoré sa síce venujú objektovo-orientovanému modelovaniu, ale prevažne v teoretickej rovine a nezohľadňujú objektovo-orientovanú databázu. Napriek tomu táto literatúra bola východiskovou pri návrhu ukladania dát do databázy. Nasledoval výber vhodného programovacieho prostredia. Ako ideálny programovací jazyk bol zvolený Smalltalk s databázou Gemstone.

Hlavným cieľom a podstatou diplomovej práce v ďalšej fáze bolo poukázať na možnosť využitia objektovo-orientovaného prístupu v geografických informačných systémoch. Porovnanie tohto prístupu s doposiaľ najviac využívaným relačným, resp. objektovo-relačným prístupom. Výhody objektovo-orientovaného prístupu eliminujú nevýhody relačného, najmä pokiaľ ide o redukciu množstva tabuliek v databáze. V objektovo-orientovanom prístupe sa nevyskytujú problémy relačných databáz typu väzobné tabuľky many:many alebo atribúty nadobúdajúce viacnásobné hodnoty v rôznych tabuľkách. Práca s objektovo-orientovanou databázou má viacero logických aj výkonových výhod. Menší počet prístupov na disk za účelom získania rôznych informácií o danom objekte. Odkazovanie sa na príbuzné objekty pomocou referencii, bez potreby zložitých dotazov alebo spájania tabuliek. Z logického hľadiska umožňuje objektovo-orientovaný prístup lepšie

pochopenie celého systému pre koncového používateľa. Takmer všetky časti reálneho sveta sú vnímané ako objekty, ktorým sa implementácia objektovo-orientovaným prístupom snaží, čo najviac priblížiť. Aj pohľad na samotnú objektovo-orientovanú databázu umožňuje používateľovi si vytvoriť lepšiu predstavu a pochopiť modelovaný systém, bez nutnosti poznania množstva tabuliek odkazujúcich sa navzájom na rôzne entity. Tento prístup uľahčuje prácu okrem koncového používateľa aj samotným programátorom a vývojárom GIS systémov. Od počiatočnej špecifikácie a analýzy, návrhu až po testovanie systému. V súčasnej dobe sa javí ako najväčší problém cena takéhoto riešenia. Objektovo-orientované databázy sú oveľa drahšie ako relačné databázy. A práve ekomonické hľadisko je jedno z najdôležitejších pri vývoji informačných systémov. Podobne ako aj „know-how“, vďaka ktorému sú momentálne objektovo-orientované GIS systémy len na začiatku svojho vývoja.

Súčasťou diplomovej práce bola aj metodická časť a prípadová štúdia venujúca sa modelovaniu objektovo-orientovaných GIS systémov. Popisuje základné analytické operácie, ktoré je možné vykonávať s vektormi aj rastrami. Vzhľadom k tomu, že v súčasnosti dominujú GIS systémy založené na relačných databázach, bola táto práca aj trošku experimentálna. Doposiaľ neexistujú žiadne štandardy, ako by sa mali geografické objekty ukladať do objektovo-orientovanej databázy, aké by mali mať atribúty, metódy, či referencie na iné objekty. Navrhnuté riešenie by malo byť základom, ktorý môže poslúžiť na jeho prípadne rozšírenie. Možnosti rozšírenie vidím najmä v oblasti implementácie, t.j. rozšírenia databázového návrhu a jeho následné implementovanie. Vychádzajúc z faktu, že skúsenosti s programovním v jazyku smalltalk boli takmer minimálne, je v tejto oblasti do budúcna veľa príležitosti na. Samotný vývoj objektovo-orientovanej metodiky sa určite v budúcnosti prejaví aj v GIS systémoch a nepochybne bude veľkým prínosom vo všetkých odvetviach, kde sa GIS systémy využívajú. Možno sa v nich prejavia aj metódy a prístupy popísané a navrhnuté v tejto diplomovej práci.

7 Literatúra

[1] BURROUGH, P.A. (1986): Principles of geographical information systems for land resources assessment, Clarendon Press (Oxford Oxfordshire and New York)

[2] WILD, J. (2003): Analýza dat v GIS. IBOT. Dostupné z WWW: http://www.ibot.cas.cz/personal/wild/data/gis_lect/gis_01_cojetoGIS.ppt (28.04.2010)

[3] JEDLIČKA, K. (2003): Úvod do GIS. Dostupné z WWW: http://gis.zcu.cz/studium/ugi/Prezentace/04-ProstorovaDataRastroveDatoveModely.pdf (4. 12. 2009)

[4] TÖRÖK, A. (2004): Objektovo-orientovaný prístup a jeho vlastnosti. Dostupný z WWW: http://www.gratex.com/Download/OOANS01Postupnost.pdf (28.12.2009)

[5] GOLDBERG, A. and D. ROBSON (1983): Smalltalk-80: The language and Its implementation. Reading, MA:Addison-Wesley Publishing Company

[6] BRODIE, M. (1984): „On the development of data models.“ In on conceptual modeling. M. Brodie, J. Milopoulos, and J.Schnidt , eds. New York, NY: Springerverlag, 19 – 48.

[7] SMITH, J. and D. SMITH (1977): Database abstractions: Aggregation and Generalization. ACM Transactions of database systems, 2(2): 105-133.

[8] DAHL, O.-J. and K. NYGAARD (1966): SIMULA – An algol-based simulation language. Communication of the ACM, 9(9): 671 – 678.

[9] MENNIS, J. L.; PEUQUET, D. J.; QIAN, L. (2000): A conceptual framework for incorporating cognitive principles into geographical database representation. Int. J. Geographical Information Science, 2000, Vol. 14, No. 6, p. 501.

[10] TANG, A.Y.; ADAMS, T.M.; USERY, E.L.(1996): A spatial data model design for feature-based geographical information systems. Int. J. Geographical Information Systems, Vol. 10, No. 5, s. 643.

[11] THOMASSON, A. L. (2001): Geographic Objects and the Science of Geography. Topoi. 2, 20, s. 149-159. Dostupný z WWW: http://www.springerlink.com/content/w1354054v2411g10/ (20.03.2010)

[12] STREIT, U. (1998): Geoinformatics (Universität Munster). Dostupný z WWW: http://ifgi.unimuenster.de/vorlesungen/Geoinformatics (28.03.2010)

[13] ROMAN, S. (1999): Microsoft Access - Návrh a programování databází. Praha (Computer Press).

[14] SHEKHAR, S., et al. (1997): Data models in geographic information systems. Communications of the ACM. 40, 4, s. 103 - 111 . Dostupný z WWW: http://portal.acm.org/citation.cfm?id=248448.248465. (01.04.2010)

[15] MITÁŠOVÁ, I., HÁJEK M. (1994): Definovanie štandardov na prenos digitálnych prostorových informácií. Kartografické listy, 2, 21-36.

[16] RAPANT, P.(1999): Úvod do geografických informačních systémů. GeoInfo, 1-3, Príloha Škola. (Computer Press).

[17] DSS Resources. (2009): Decision Support Systems Resources. Dostupné z WWW: http://www.dssresources.com/. (13.04.2010)

[18] LEVČÍKOVÁ, A. (2007): Ministerstvo obrany SR. 2007 Topografický ústav Banská Bystrica. Dostupné z WWW: <http://topu.army.sk/aktivity/zbornik02/13_levcikova.pdf>. (14.04.2010)

[19] Žilinský samosprávny kraj [online]. 2010. GIS ŽSK. Dostupné z WWW: <http://www.zask.sk/showdoc.do?docid=8214>. (14.04.2010)

[20] FERKOVÁ, L. AND MARŠÍK, V. (2009): Využití GIS v operačním řízení HZS GISelIZS AE v praxi. GIS Ostrava 2009 [online]. Dostupné z WWW: <www.tmapy.cz/docs/aktualne/clanky/gisostrava_giselIZSAE.pdf>. (28.04.2010)

[21] http://www.smalltalk.org (3.1.2010)

[22] ESRI: ESRI Shapefile Technical Description [online]. United States of America : Environmental Systems Research Institute, Inc., 1998 [cit. 2010-05-11]. Dostupné z WWW: <http://www.esri.com/library/whitepapers/pdfs/shapefile.pdf>.

[23] MapCruzin [online]. 1996. Free Mexico Canada U.S. Transportation Shapefiles. Dostupné z WWW: <http://www.mapcruzin.com/download-mexico-canada-us-transportaton-shapefile.htm>. (19.05.2010)

Zoznam príloh

Príloha 1. Definície tried objektovo-orientovaných vektorov a rastrov v databáze Gemstone.

Príloha 2. CD médium obsahuje elektronickú verziu technickej správy diplomovej práce vo formátoch .doc a .pdf. CD obsahuje aj image aplikácie v jazyku smalltalk.

Príloha 1

Trieda Point:

```
Object subclass: 'Point'
  instVarNames: #( coordX coordY pointType
                   startNode endNode pointBelongsTo)
  classVars: #()
  classInstVars: #()
  poolDictionaries: #[]
  inDictionary: UserGlobals
  constraints: #[ #[ #coordX, Integer],
                  #[ #coordY, Integer],
                  #[ #pointType, String],
                  #[ #startNode, Boolean],
                  #[ #endNode, Boolean] ]
  instancesInvariant: false
  isModifiable: true
```

Trieda Line:

```
Point subclass: 'Line'
  instVarNames: #( startPoint endPoint leftNeighbour
                   rightNeighbour lineBelongsTo lineLength crossObject)
  classVars: #()
  classInstVars: #()
  poolDictionaries: #[]
  inDictionary: UserGlobals
  constraints: #[ #[ #startPoint, Point],
                  #[ #endPoint, Point],
                  #[ #leftNeighbour, Polygon],
                  #[ #rightNeighbour, Polygon],
                  #[ #lineLength, Integer] ]
  instancesInvariant: false
  isModifiable: true
```

Trieda PolyLine:

```
Line subclass: 'PolyLine'
  instVarNames: #( lineNumber lines crossObjects)
  classVars: #()
  classInstVars: #()
  poolDictionaries: #[]
  inDictionary: UserGlobals
  constraints: #[ #[ #lineNumber, Integer],
                  #[ #lines, Line] ]
  instancesInvariant: false
  isModifiable: true
```

Trieda Polygon:

```
Line subclass: 'Polygon'
  instVarNames: #( lines startEndNode perimeter
                   centroid)
  classVars: #()
  classInstVars: #()
  poolDictionaries: #[]
  inDictionary: UserGlobals
  constraints: #[ #[ #lines, Line],
                  #[ #startEndNode, Point],
                  #[ #perimeter, Integer],
                  #[ #centroid, Integer] ]
  instancesInvariant: false
  isModifiable: true
```

Trieda Raster:

```
Object subclass: 'Raster'
  instVarNames: #( name rasterCell rasterLegend)
  classVars: #()
  classInstVars: #()
  poolDictionaries: #[]
  inDictionary: UserGlobals
  constraints: #[ #[ #name, String],
                  #[ #rasterCell, Array],
                  #[ #rasterLegend, Array] ]
  instancesInvariant: false
  isModifiable: true
```