

STAVEBNÍ FAKULTA ČVUT
KATEDRA MAPOVÁNÍ A KARTOGRAFIE



DIPLOMOVÁ PRÁCE

Jan Kohout:

Tvorba webové mapové aplikace pro vizualizaci zájmových
objektů Českého horolezeckého svazu

16.5.2008

PROHLÁŠENÍ

PROHLÁŠENÍ

Prohlašuji, že jsem diplomovou práci vypracoval samostatně a že byly uvedeny veškeré zdroje informací, ze kterých jsem čerpal.

V Jihlavě, 16.5.2008

Jan Kohout

PODĚKOVÁNÍ

PODĚKOVÁNÍ

Na tomto místě bych rád poděkoval vedoucímu diplomové práce Ing. Jiřímu Cajthamlovi, Ph.D. za věcné připomínky a vstřícný přístup při vedení práce.

ABSTRAKT

Tato diplomová práce je zaměřena na tvorbu mapové služby v prostředí JavaScript sloužící pro potřeby webu Skalních oblastí Českého horolezeckého svazu (ČHS). Práce představuje veřejné API služby a na základě stanovených kritérií je určeno konečné řešení, na kterém bude služba postavena. Dále je řešen způsob připojení vstupních dat ČHS z databáze Oracle pomocí ASP skriptů a vizualizace zájmových bodů v mapě. Současně je řešena koncepce vzhledu uživatelského prostředí, návrh ovládacích prvků a implementace funkcí vyhledávání. Výsledkem diplomové práce je interaktivní mapová aplikace provozovaná na webu ČHS.

Klíčová slova:

Google maps API, amapy.cz API, mapy.cz API, JavaScript, interaktivní mapa, POI, ČHS, databáze Oracle

ABSTRACT

This diploma thesis is focused on creating of a map application service using JavaScript language for needs of the Rock regions web page of the Czech Mountaineering Association (ČHS). Public API services are summarized and on the basis of the established criteria the final solution is chosen. On this solution the service will be created. Consequently, visualization of the points of interest is being solved and the connection method of input data of ČHS from Oracle database is resolved by ASP scripting. One part is dealing with solving the approach of a client workspace, design of control elements and implementation of search functions. The result of this diploma thesis is an interactive web map application which runs on the web of ČHS.

Keywords:

Google maps API, amapy.cz API, mapy.cz API, JavaScript, interactive map, POI, ČHS, Oracle database

OBSAH

Obsah	i
Seznam obrázků	iv
Seznam tabulek	v
Seznam zkratek	vi
1 Cíl práce	1
1.1 Požadavky na aplikaci	1
1.1.1 Požadavky obsahové	2
1.1.2 Požadavky formální a technologické	2
1.2 Současný stav	2
2 Představení řešení vhodných k realizaci služby	4
2.1 Mapové servery	4
2.2 Mapové služby s API rozhraním	5
2.3 Google Maps API [url 3]	6
2.3.1 Charakteristika	6
2.3.2 Vlastnosti a funkce	6
2.3.3 Mapové podklady	7
2.3.4 API a dokumentace	7
2.3.5 Podmínky užití API	7
2.4 mapy.cz API	8
2.4.1 Charakteristika	8
2.4.2 Vlastnosti a funkce	8
2.4.3 Mapové podklady	9
2.4.4 API a dokumentace	9
2.4.5 Podmínky užití API	9
2.5 amapy.cz API	10
2.5.1 Charakteristika	10
2.5.2 Vlastnosti a funkce	10
2.5.3 Mapové podklady	10
2.5.4 API a dokumentace	10

2.5.5	Podmínky užití API	11
2.6	Zhodnocení výběru technologie.....	11
3	Představení použitých technologií (teoretické)	13
3.1	Databázová komponenta – server Skalních oblastí ČR.....	13
3.1.1	Databázový systém Oracle10g Express Edition	14
3.1.2	ASP	14
3.1.3	SQL.....	15
3.2	Mapová komponenta.....	16
3.2.1	API amapy.cz založené na JavaScript a MooTools framework	16
3.2.2	Souřadnicové systémy podporované API.....	17
3.2.3	HTML a CSS	18
4	Realizace mapového modulu.....	19
4.1	Principy tvorby mapy.....	21
4.2	Sestavení <i>Recordset</i>	23
4.3	Navigační a kompoziční prvky	24
4.3.1	Navigační panel	24
4.3.2	Grafické měřítko	24
4.3.3	Přepínače vrstev	25
4.3.4	Informační bublina.....	26
4.4	POI a elastická bublina	26
4.4.1	Záložky bubliny	27
4.4.2	Patička bubliny a její funkce.....	28
4.4.3	Okno maximalizace	29
4.5	Informační panel – <i>Sidebar</i>	31
4.5.1	Obsah panelu.....	31
4.5.2	Funkce panelu	33
4.6	Kontextové menu a jeho funkce	34
4.7	Vyhledávací panel.....	35
4.7.1	Nabídka – formulář typu <i>Select</i>	35
4.7.2	Vyhledávání – formulářové pole typu <i>Input</i>	36
4.7.3	Tlačítka <i>smazat</i> a <i>vyhledat</i>	38
5	Závěr	40

6	Zdroje informací.....	42
6.1	Publikace.....	42
6.2	Webové stránky a články	43
6.3	Originální dokumentace v elektronické podobě	43
7	Přílohy.....	44
7.1	Obsah přiloženého CD.....	44

SEZNAM OBRÁZKŮ

2.1:	Schéma funkce mapového serveru [1].....	5
2.2:	Ukázka využití Google API - statistika prezidentských primárek v Kentucky	8
3.1:	Diagram relací tabulek databáze použitých pro realizaci mapové aplikace	14
3.2:	Na pravé straně je zobrazen kód ASP před požadavkem klienta na jeho zobrazení. Pravá strana obsahuje kód, tak jak jej obdrží prohlížeč.	15
3.3:	Takto se kód zobrazí v prohlížeči	15
4.1:	Prostředí programu Firebug integrovaného do okna prohlížeče – prověření objektu	21
4.2:	Ukázka minimálního kódu nutného pro vykreslení mapy	22
4.3:	Sestavení SQL dotazu s využitím příkazu replace.....	24
4.4:	Přidání navigačního panelu do mapy	24
4.5:	Grafické měřítko	25
4.6:	Přidání mapových prvků spravující mapové vrstvy	26
4.7:	Přepínače typů map a překryvných vrstev	26
4.8:	Informační bublina.....	26
4.9:	Deklarace proměnné marker typu AMarker	27
4.10:	Definice záložek a patičky bubliny	28
4.11:	Maximalizovaná bublina s podrobným výpisem skály.....	30
4.12:	ASP skript pro tvorbu seznamu regionů v panelu Sidebar	31
4.14:	Informační panel - výběr oblasti	33
4.15:	Funkce createJumps() - tvoří položky menu a proměnné s parametry polohy a přiblížení	34
4.16:	Kontextové menu – poblíž menu je viditelný kurzor, ke kterému jsou vztaženy zobrazené souřadnice	34
4.17:	Okno pro vyhledávání skal	35
4.18:	Generování formuláře typu Select pomocí ASP skriptu.....	36
4.19:	Funkce zoomChoice().....	36
4.20:	Sestavení SQL dotazu v souboru <i>find.asp</i>	37
4.21:	Okno vyhledávání s "našeptávačem"	38

SEZNAM TABULEK

2.1:	Srovnání mapových služeb	11
4.1:	Přehled recordset záznamů použitých pro tvorbu mapového obsahu.....	23

SEZNAM ZKRATEK

API – *Application Programming Interface*, rozhraní s dostupnou dokumentací pro možnosti veřejného užití za současného splnění licenčních podmínek

ASP – *Active Server Pages*, technologie umožňující vykonávání kódu na straně serveru (server-side)

CGI – *Common Gateway Interface*, protokol zajišťující komunikaci mezi webovým serverem a serverovými aplikacemi

CSS – *Cascading Style Sheets*, formát definující vzhled webového obsahu

ČHS – *Český horolezecký svaz*

ČÚZK – *Český úřad zeměměřický a katastrální*

DHTML – *Dynamic HTML*, soubor technologií ke tvorbě interaktivních stránek

DOM – *Document Object Model*, rozhraní pro dynamický přístup a aktualizaci obsahu i formy HTML dokumentů

ECMA – *European Computer Manufacturers Association*,
<http://www.ecma-international.org/>

FTP – *File Transfer Protocol*, protokol pro vzdálený přenos souborů

GIS – *Geographic Information System*, systém zabývající se prostorově určenými daty

GPS – *Global Positioning System*, veřejně dostupný navigační systém

HTML – *HyperText Markup Language*, značkovací jazyk používaný na webu

HTTP – *Hypertext Transfer Protocol*, protokol nejčastěji využívaný pro přenos HTML dokumentů

IIS – *Internet Information Services*, webový server společnosti Microsoft

ISO – *International Organization for Standardization*, <http://www.iso.org>

IWM – *Interactive Web Maps*, interaktivní webové mapy

JS – *JavaScript*, objektově orientovaný programovací jazyk

JSON – *JavaScript Object Notation*, formát pro výměnu dat

-
- OGC – *Open Geospatial Consortium*, organizace zaměřená na standardy týkající se GIS
- PHP – *Hypertext Preprocessor*, skriptovací programovací jazyk (server-side) pro tvorbu dynamických stránek
- POI – *Points of Interest*, zájmové body
- SQL – *Structured Query Language*, dotazovací jazyk relačních databází
- SVG – *Scalable Vector Graphics*, jazyk pro popis 2D grafických objektů a aplikací v XML[1]
- URL – *Unique Resource Locator*, způsob, jak jednoznačně zapsat umístění souboru na internetu/intranetu [url 1]
- VML – *Vector Markup Language*, jazyk z rodiny XML pro popis vektorové grafiky
- W3C – *World Wide Web Consortium*, vyvíjí a definuje webové technologie
- WGS84 – *World Geodetic System*, definuje referenční plochu Země
- WMS – *Web Map Service*, služba vyvinutá OGC pro distribuci a sdílení GIS dat v prostředí internetu
- WWW – *World Wide Web*, aplikační vrstva protokolu HTTP

1 CÍL PRÁCE

Cílem práce je implementovat mapový modul do stávající podoby webu Skalních oblastí ČR a vyřešit propojení a vizualizaci databázových informací na mapě. Nutnou podmínkou vizualizace je informace o prostorovém určení zájmového bodu (dále POI), které je v podobě souřadnic systému WGS84 součástí databáze. Dalším úkolem je výběr relevantních informací, které se uživateli zobrazí v informační bublině po výběru daného POI. Aplikace musí být vybavena funkcí vyhledávání v zobrazených POI a také možností zobrazení skupiny POI se společnými znaky, jako je příslušnost k té které skalní oblasti, případně regionu.

1.1 Požadavky na aplikaci

Hlavní motivací práce je tvorba klientské mapové služby, která se stane nedílnou součástí webu Skalních oblastí ČR a nosným modulem řešícím vizualizace rozsáhlých databázových informací v mapě. Výsledný modul bude volně dostupný na internetu (bez nutnosti registrace a přihlášení). Klientem se rozumí návštěvník, který má právo na prohlížení obsahu a disponování všemi funkcemi v mapě užitými. Profil uživatele je unikátní a jednotný, to znamená že nebude řešena tvorba účtu s nadřazenými uživatelskými právy (administrátor).

Pro samotnou přívětivost aplikace je nutné definovat co návštěvník od aplikace očekává po stránce obsahové i formální a stanovit jaké funkce mu zpříjemní komfort v ovládání.

1.1.1 Požadavky obsahové

Vzhledem k faktu, že mapový modul má za cíl zjednodušit a zpřehlednit orientaci uživatele v leckdy nepřehledných a špatně přístupných terénech, je prioritou kvalita mapových dat. Tím se rozumí rozlišení mapového podkladu pro ČR alespoň v měřítku 1:10 000 zahrnující letecké snímky a v nejlepším případě i překryvné vrstvy značených turistických cest a cyklostezek.

Součástí modulu by mělo být také kvalitní uživatelské rozhraní, zahrnující následující prvky tvořící obsah mapové aplikace:

- navigační – posun mapy, ovladač změny měřítka
- kompoziční – grafické měřítko, případně legenda
- obslužné – přepínače vrstev, vyhledávání v POI, odečítání souřadnic

Předchozí výčet zahrnuje pouze složky tvořící samotný obsah mapy, dále je proto nutné vymezit i požadavky na technologie a způsob zpracování aplikace:

1.1.2 Požadavky formální a technologické

Následující výčet zvažuje a zahrnuje složky vymezující technologický a formální rámec mapové aplikace založené na dostupném API:

- freeware/OpenSource¹ řešení
- volně dostupné mapové podklady
- propojení s databází Oracle
- nenáročnost implementace a údržby aplikace
- stabilita

1.2 Současný stav

Do doby implementace mapové aplikace je řešení vizualizace POI na webu Skalních oblastí ČR² řešeno přes URL odkaz, ve kterém jsou přeneseny informace o

¹ Počítačový software s volně dostupným veřejným kódem a s licencí umožňující tento kód volně užívat.

poloze (GPS) jako parametry za vedoucím znakem # (hash)³. Ten se generuje ASP skriptem. Tímto způsobem je možné velice jednoduše zobrazit danou skálu například na mapových službách: *maps.google.com*, *mapy.cz* a také na *amapy.cz*. Přestože je možné předat informace o více POI, je velmi problematické řešení způsobu jejich zobrazení. Informace se skládají do jedné url, ale není možné jakkoliv ovlivnit nastavení zobrazení a případně i ovládání mapy zobrazené stránky. Další nepříjemností je zobrazení POI na doméně mapového serveru a nemožnost integrace mapového prostředí přímo do struktury konkrétního webu. Veškeré tyto nevýhody se ale dají eliminovat využitím mapového API příslušných webů, což je hlavní náplní práce.

² <http://skaly.horosvaz.cz/>

³ Cílová stránka přebere parametry z url a na jejich základě vygeneruje výsledek.

2 PŘEDSTAVENÍ ŘEŠENÍ VHODNÝCH K REALIZACI SLUŽBY

2.1 Mapové servery

Přestože tato řešení nejsou předmětem naplňujícím zadání diplomové práce (nenabízí své vlastní API), jsou natolik významná a rozšířená, že považuji za správné je zde zmínit.

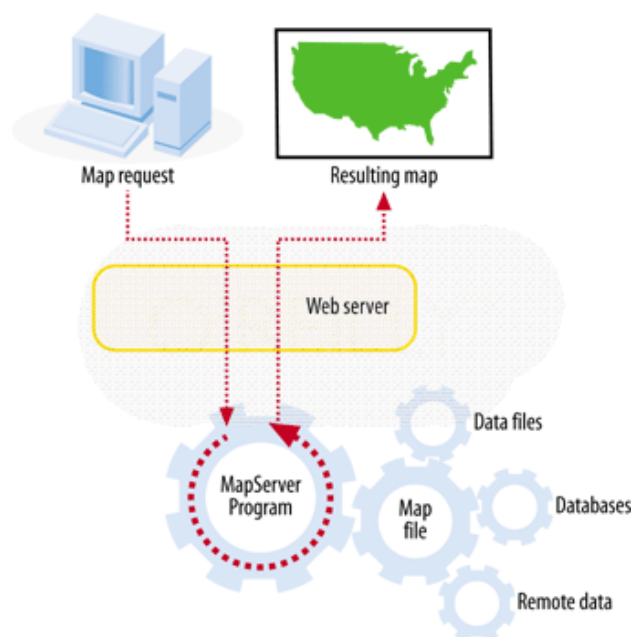
Předností těchto serverů je především univerzálnost použití. Vlastní aplikaci je možné vytvořit přesně na míru užitím specifických mapových podkladů a prvků mapy (legenda, grafické měřítko, navigace). Tvůrce není omezen použitou platformou (Windows, Linux, Mac OS, Solaris, atd.), ani skriptovacím a vývojovým prostředím (PHP, JavaScript, VBScript, Python, atp.).

Jedno z možných technologických řešení je užití nekomerčního mapového serveru jako je UMN MapServer vyvíjený Minnesotskou universitou. Ten spadá do kategorie IWM služeb, pracuje jako CGI program a zprostředkovává spojení mezi datovým a webovým serverem. Podporována je široká škála skriptovacích jazyků. Tento způsob komunikace nabízí možnost využít pokročilé webové nástroje jako DOM, CSS. Pro použití UMN MapServeru hovoří i široká komunita uživatelů, kvalitní dokumentace a hlavně srovnatelnost s celou řadou komerčních produktů jako jsou ArcIMS, Geomedia Web Map, MapGuide, apod.

Jistou nevýhodou je ale v případě užití tohoto řešení zdroj mapových podkladů. Pokud chceme vyhovět podmínce jejich volné dostupnosti, v úvahu připadá užití dat distribuovaných pomocí WMS. Data v této podobě jsou k dispozici například na Portálu veřejné správy⁴, případně na webu ČÚZK⁵.

⁴ <http://geoportal.cenia.cz/mapmaker/cenia/portal/help/cz/sluzby.html>

Nevýhodou takto poskytovaných dat je zpravidla omezený měřítkový rozsah i samotný obsah mapy. Dalším negativem je samotná vývojová náročnost, kterou je možné alespoň z části eliminovat užitím standardizovaných funkcí, které jsou vývojářům k dispozici. Vývojář se ale neobejde bez zběžné znalostí kartografie a problematiky souřadnicových systémů a jejich projekcí.



Obr. 2.1: Schéma funkce mapového serveru [1]

2.2 Mapové služby s API rozhraním

Všechny následující služby mají společné použití moderních technologií charakteristických pro web 2.0⁶. Pro vývojáře znamenají na jedné straně usnadnění vzniklé odstíněním problematiky implementace mapových podkladů a částečně i práce s nimi, na straně druhé ale kladou větší nároky na znalost nejnovějších JavaScript framework⁷, mezi kterými konvergují. Přestože většinou umožňují užití vlastních mapových podkladů, pro běžnou potřebu si uživatel s těmi obsaženými vystačí.

Pro každý stupeň přiblížení je definován různý mapový podklad. Ten je rozřezán na dílce a ty jsou ve formě jednotlivých obrázků zasílány prohlížeči, který z nich sestaví

⁵ http://www.cuzk.cz/Dokument.aspx?PRARESKOD=998&MENUID=0&AKCE=DOC:10-WMS_PRO_KM

⁶ Poněkud nejednoznačný termín používaný v kontextu technologií vzniklých přibližně v průběhu posledních čtyř pěti let. Jeho zavedení bylo nejspíše motivováno snahou odlišit statický web od dynamických internetových stránek.

⁷ Jedná se o souhrn softwarových řešení, které je možné převzít a používat ve svých projektech.

obraz požadované oblasti. Z těchto principů vyplývá, že uživatel vždy obdrží pouze mapu, která je dostupná v konkrétním měřítku. Na rozdíl od mapových serverů založených na podobných principech jako jsou ty zmíněné v předchozím odstavci, které vždy na základě požadavku klientského prohlížeče zpět odešlou unikátní výsledek dotazu, jež je sestaven z prvků obsažených v databázi.

Následující text se soustřeďuje na tři hlavní interaktivní mapové služby disponujícími API a jejich zevrubným popisem:

1. *maps.google.com*,
2. *mapy.cz*,
3. *amapy.cz*

2.3 Google Maps API [url 3]

2.3.1 Charakteristika

Google Maps, uveřejněné společností Google v únoru 2005, co se webových mapových služeb týká, je nejstarší a také nejrozšířenější „hráč“ na trhu těchto aplikací. Jeho přívětivé uživatelské prostředí zahrnující možnosti posunu mapy pomocí metody *drag&drop*⁸, změna měřítka mapového podkladu pomocí kolečka myši a přepínání mapových vrstev z klasické mapy na satelitní či ortofotomapu se stalo standardem i inspirací pro všechny pozdější konkurenty.

2.3.2 Vlastnosti a funkce

Stěžejní funkcí pro rychlou orientaci v mapě je vyhledávání, kterému se v případě požadavku na objekt mapy říká *geocoding* (*geocode*)⁹. V případě hledání českých adres a služeb se projevuje globální zaměření aplikace a ta nepodává zdaleka tak dobré výsledky, jako je tomu při požadavcích zájmových bodů například v USA.

Jak je popsáno výše, prvky ovládající nastavení přiblížení a posun mapového podkladu jsou nezbytně nutné pro práci a orientaci v mapě. Nabídka módů zobrazení je poměrně pestrá. Rozlišuje pohled *Map* (topografické mapy a plány měst a obcí),

⁸ „Uchop a táhni“. Při stisku levého tlačítka myši je možné pohybovat podkladem.

⁹ Prvek databáze k identifikování polohy požadavku a následnému zobrazení v mapě.
[<http://www.digimap.gg/glossary>]

Satellite (satelitní a letecké snímky) a jako poslední volbu *Terrain* (geomorfologické vlastnosti). Módy *Street View* a *Traffic* jsou v případě ČR prozatím bez využití.

2.3.3 Mapové podklady

V současné době mapové podklady pokrývají prakticky celý svět. S detailností a kvalitou dat, konkrétně pro Českou republiku, už to tak slavné není. V případě leteckých snímků je situace v rámci ČR značně rozpačitá. Praha a další velká města jsou pokryta srovnatelně, jako je tomu u českých služeb, ale detailnost dat mimo ně je takřka nedostačující.

V případě topografických map je situace stejná. Proto ani chybějící vrstvy s možností zobrazení turistických tras, či cyklostezek nepřekvapí.

2.3.4 API a dokumentace

Jedno z největších pozitiv v případě užití Google Maps API pro tvorbu vlastních *mashup*¹⁰ aplikací je rozsáhlá a aktuální dokumentace a nepřeborné množství internetových diskusí a webových stránek zaměřených na tuto problematiku. I z tohoto důvodu je zajištěna kvalitní podpora a obrovské množství uživatelů je zárukou bezproblémového provozu na všech dostupných platformách.

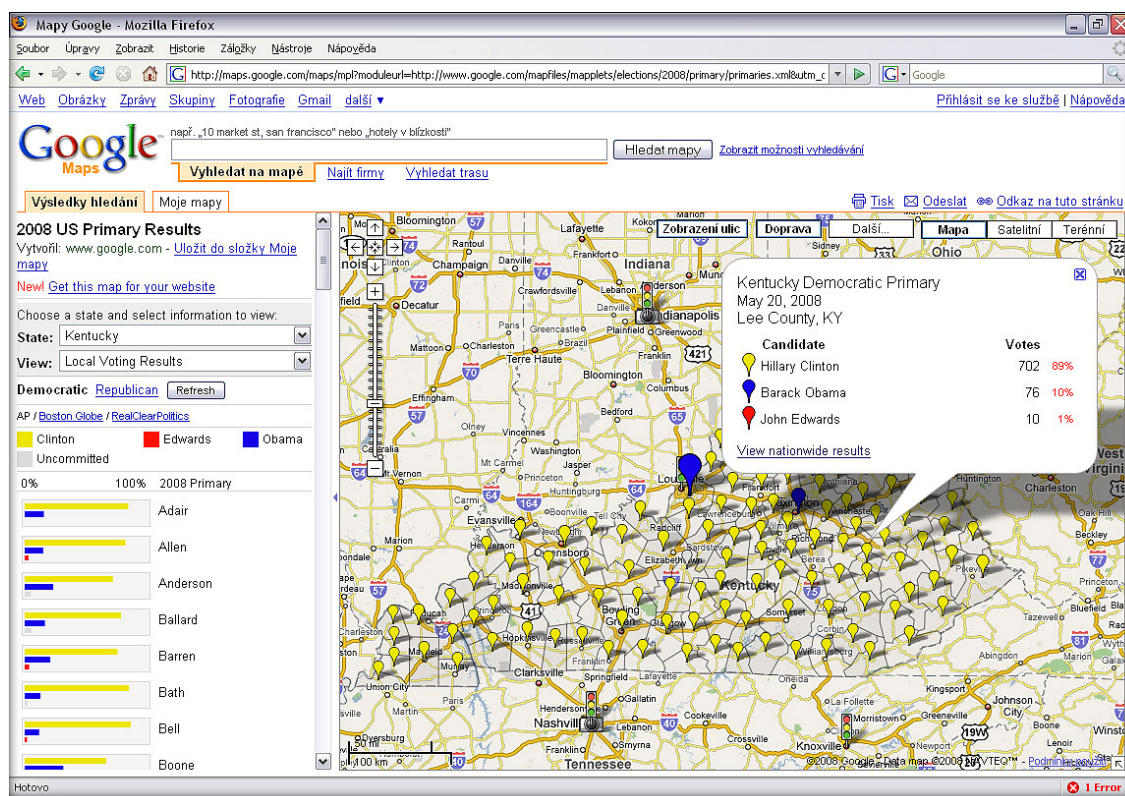
2.3.5 Podmínky užití API

Pro užití API je nutné vytvoření Google účtu a vygenerování unikátního klíče pro doménu, kde se bude mapová služba nacházet. Vytvořený Google účet ale nemá omezený počet domén, takže je možné pod jedním spravovat více mapových aplikací.

Hlavní a základní rysy podmínek jsou tyto [url 2]:

- Počet zobrazených stránek během dne je bez omezení. (V případě překročení hranice 500 000 přístupů denně je po konzultaci možné navýšení konektivity ze strany Google)
- Limit 15 000 požadavků na geocode (24 hodinová perioda)
- Výsledná služba musí být volně dostupná koncovému uživateli

¹⁰ mashup = kombinace dat a funkcí z více zdrojů pro tvorbu webové aplikace.



Obr.2.2: Ukázka využití Google API - statistika prezidentských primárek státu Kentucky

2.4 mapy.cz API

2.4.1 Charakteristika

Průkopník na českém internetu v počátcích obdivovaný a hojně používaný s příchodem dalších konkurentů začíná lehce ztrácet svůj dřívější náskok. Přesto si „dítě“ společnosti *seznam.cz* zachovává i nadále významnou pozici na trhu.

2.4.2 Vlastnosti a funkce

Od svého vzoru – Google Maps – přebírá většinu dobře fungujících vlastností, ovládacích prvků a není se čemu divit. V základním zobrazení nechybí panel ovládání přiblížení a posunu, přepínače vrstev, grafické měřítko a najdeme zde i ikony sloužící pro odečet souřadnic (WGS84) z mapy a nástroj pro měření vzdáleností.

Vyhledávací schopnosti v rámci ČR jsou na velmi dobré úrovni a stejně tak si vede i funkce *Plánovač trasy* s možností upřesnění dotazu pomocí parametrů. K výběru slouží následující volby: *Nejrychlejší/Nejkratší cesta, Placené/Neplacené úseky*.

2.4.3 Mapové podklady

V případě *Základního pohledu* mapa obsahuje zjednodušenou dopravní síť, zástavbu a porosty. V závislosti na stupni přiblížení se zobrazují i zájmové body (zastávky, parkoviště, divadla, banky,...). Mapové podklady jsou aktualizované dvakrát ročně. Další možností je mapa *Turistická*, založená na turistických mapách společnosti SHOCart. Obsahuje vypínatelné podvrstvy *Stínování* (pro údajnou lepší představu o reliéfu), *Turistické trasy* a *Cyklostezky*. Stejně tak je tomu i v případě volby módu *Fotomapa*.

2.4.4 API a dokumentace

Časté problémy s přechodem na novější verzi, řada nedokumentovaných funkcí a mizivá podpora ze strany tvůrců jsou základ pro odmítavý postoj většiny potenciálních uživatelů.

2.4.5 Podmínky užití API

Stejně jako je tomu u Google, je nutné pro adresu www stránky generovat klíč a potvrdit souhlasem podmínky užití API. Nejslabším článkem celé mapové služby jsou podmínky licenčního ujednání¹¹ a jeho tři nosných pilířů:

- Mapy API lze použít pouze pro nekomerční provoz.
- Mapy API má k dispozici omezené mapové podklady oproti službě *mapy.cz*.
- Mapy API má omezený denní počet zobrazení (v současné době 1000 denně).

Při porovnání s ustanovením Google Maps je na první pohled patrné, proč se API *mapy.cz* stává prakticky nepoužitelné. Nekomerčnost užití a omezenost mapových podkladů je snad ještě akceptovatelné a není tak omezující, jako samotná podmínka pouze tisíce přístupů denně.

¹¹ <http://api.mapy.cz/keygen>

2.5 amapy.cz API

2.5.1 Charakteristika

Dlouho byly *amapy.cz* společnosti *atlas.cz* ve stínu *mapy.cz*. V posledních měsících ale nabírá tento web na kvalitě a snaží se nabízet více uživatelského komfortu a služeb i v podobě API rozhraní. Současné srovnání už začíná vyznívat spíše pro volbu tohoto hráče.

2.5.2 Vlastnosti a funkce

Není překvapivé, že ani *amapy.cz* se neodchylují od pomyslného standardu a v hlavním okně najdeme vše obvyklé a na svém místě. Z tohoto hlediska rozhodují spíše subjektivní sympatie. Nepatrným avšak bezvýznamným odlišením je možnost volně přesunovat panel navigace.

2.5.3 Mapové podklady

Ani v této kategorii se nesetkáme s ničím novým. Tradiční vzhled je podporován i volbou zobrazovacích módů pro zobrazení *Základní*, anebo *Letecké* mapy s možností zobrazení překryvných podvrstev *Turistické trasy* a *Cyklotrasy*. Za zmínku stojí také *Vodácké mapy*.

Užitím již implementovaných podkladů ale nejsme omezeni. API nabízí možnost použití podkladů vlastních. Podporovány jsou soubory typu *jpg*, *gif*, nebo *png*. Ty je nutné „rozřezat“ na jednotlivé čtverce, které jsou vztaženy ke konkrétnímu měřítku. Pro tuto činnost je vhodné použít například software *Google Maps Image Cutter*¹².

2.5.4 API a dokumentace

API je chloubou *amapy.cz* a stejně tak jejich hlavního architekta Daniela Steigerwalda. Pravidelné aktualizace, přehledná dokumentace¹³ doplněná o řadu inspirativních příkladů, tvůrci moderované internetové diskuse¹⁴ tvoří velice pevnou základnu, na které mohou tvůrci mashup stavět svoje projekty. O kvalitách svědčí i řada již funkčních realizací.

¹² <http://www.casa.ucl.ac.uk/software/googlemapimagecutter.asp>

¹³ <http://amapy.atlas.cz/api/>

¹⁴ <http://pokec.atlas.cz/diskuze/diskuze.aspx?cat=2504>

2.5.5 Podmínky užití API

Ani v tomto případě se uživatel neobejde bez registrace webové adresy s mapovou službou a odsouhlasením podmínek užití po kterých obdrží unikátní klíč vztažený k registrované URL. Smluvní podmínky nejsou ale zdaleka tak restriktivní, jako u *mapy.cz*.

Nejdůležitější body ustanovení jsou tyto¹⁵:

- uživatelé internetu nebudou omezeni v přístupu na uživatelské stránky, kde je Služba použita
- užití Služby na uživatelských stránkách bude zcela zdarma a bez komerčního záměru

			
podrobnost leteckých snímků*	dobrá	výborná	výborná
kvalita mapového podkladu*	dobrá	výborná	výborná
podrobnost plánů itavilánu	velmi dobrá	výborná	výborná
turistické mapy/cyklostezky	ne/ne	ano/ano	ano/ano
zájmové body - podrobnost	nedostatečná	výborná	výborná
kvalita dokumentace API	výborná	dobrá	výborná
aktualizace API	častá	občasná	častá
propracovanost API	výborná	Dobrá	Výborná
vlastní mapový podklad	ano	ne	Ano
Licenční omezení	15 000 přístupů/24h	1 000 přístupů/24h, nekomerčnost záměru	Nekomerčnost záměru

Tab. 2.1: Srovnání mapových služeb

2.6 Zhodnocení výběru technologie

Na základě srovnání hodnocení služeb dle Tab. 2.1 se na první pohled nedá jednoznačně vyřknout, která služba je tou nejlepší volbou. Nicméně je zřetelné, že řešení založené na API *mapy.cz* poněkud ztrácí dech právě díky svému API. Kvalita a propracovanost map je výborná, ale situaci nezachrání a stanovené licenční podmínky realizaci prakticky znemožní.

Dá se říci, že *Google Maps* jsou na tom přesně opačně. Obrovská komunita uživatelů zajišťuje časté aktualizace verzí, kvalitu a propracovanost kódu, ale v ohledu na mapové podklady se stále nedaří uspokojivě pokrýt menší, avšak významné regiony

¹⁵ <http://amapy.atlas.cz/api/licence.aspx>

jako je právě střední Evropa. Přesné informace o kvalitě a termínu obnovení mapových podkladů jsou nezjistitelné, ačkoliv jsou velmi zajímavé a leckterému potenciálnímu uživateli by pomohly. Stejně tak se „tají“ i informace o rozlišení map a leckdy i údaj o čase jejich poslední aktualizace.

Skoro se zdá, že se *amapy.cz* poučily z chyb svých konkurentů a inspirovaly se tím lepším z obou předešle komentovaných služeb. Mapový podklad je velmi kvalitní a nápaditost programátorů se kladně projevila na řadě funkcí a přehlednosti samotného API.

Vzhledem k faktu, že vzájemným srovnáním služeb vychází jako nejvšestrannější řešení API *amapy.cz* a dále toto řešení splňuje veškeré požadavky stanovené v odstavci 1.1, bude vývoj mapového modulu založen právě na tomto řešení.

3 PŘEDSTAVENÍ POUŽITÝCH TECHNOLOGIÍ (TEORETICKÉ)

3.1 Databázová komponenta – server Skalních oblastí ČR

Server Skalních oblastí ČR (SO) je postaven na platformě MS Server 2003 a webovém serveru IIS 6.0. Je navržen jako systém plně dynamický, tj. obsah (databázový systém Oracle 10g) je zcela oddělen od jeho formy – zdrojových kódů. Ty jsou napsány v jazycích ASP a ASP.Net. Díky těmto vlastnostem se každý jedinečný požadavek (request) na zobrazení stránky vyhodnotí a na straně serveru se sestaví odpověď (response), která se odešle zpět klientovi.

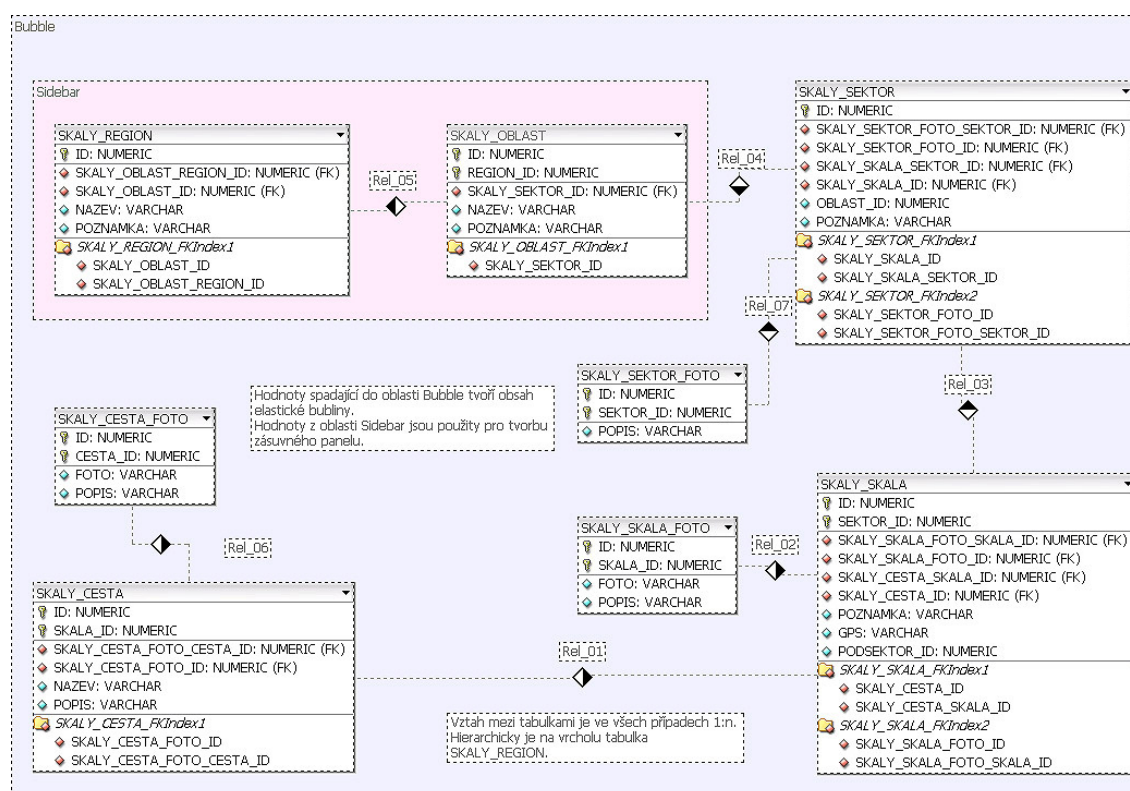
Hierarchický model databáze je dobře viditelný na Obr.3.1. Zobrazuje ale pouze tabulky, které byly použity pro zisk dat tvořící obsah mapy. Databáze jako taková je rozsáhlejší a není důvod uvádět kompletní větvení. Vztahy mezi tabulkami jsou vždy v poměru $1:n$. Jako konkrétní příklad by se dal uvést vztah, kdy na jeden záznam skály (tabulka *SKALY_SKALA*) připadá n záznamů cest na skálu vedoucích (tabulka *SKALY_CESTA*). Vazby na Obr.3.1 značené *rel_01* až *rel_07* ale nejsou tzv. pravé relační vztahy, kdy má vazba mezi tabulkami nastaveny konkrétní vlastnosti typu: „*při smazání záznamu skály smaž i tabulku zahrnující cesty na skálu vedoucí*“. V případě Skalních oblastí jsou tabulky ponechány izolovaně a vztahy mezi nimi jsou ošetřeny kódem. Značení vztahů v diagramu na Obr.3.1 je tedy čistě informativní.

Co se kódování týče, pro zajištění kompatibility a správnosti zobrazení diakritiky českého jazyka je v databázi použita znaková sada UTF-8¹⁶.

¹⁶ UTF-8=UCS Transformation Format, varianta kódování Unicode znaků.

3.1.1 Databázový systém Oracle10g Express Edition

Jedná se o objektově orientovanou relační databázi typu klient/server využívající dotazovací jazyk PL/SQL. Tento systém řízení báze dat je poskytován zdarma a to i pro komerční nasazení. Oproti vyšším (placeným) verzím má jisté funkční restrikce a stejně tak je omezena i maximální velikost spravovaných dat a využitelný výpočetní výkon serveru¹⁷.



Obr.3.1: Diagram relací tabulek databáze použitých pro realizaci mapové aplikace

3.1.2 ASP

„ASP je technologie umožňující pomocí skriptů vykonávaných na straně serveru dynamicky vytvářet obsah, který není závislý na použitém prohlížeči. Kód skriptu lze zapsat v libovolném z několika jazyků¹⁸. Skript se do jinak normálního kódu HTML, který tvoří obsah stránky, vkládá pomocí speciálních značek. Takovou heterogenní stránku se skriptem a obsahem interpretuje webový server až v okamžiku, kdy jí chce uživatel zobrazit.“ [3]

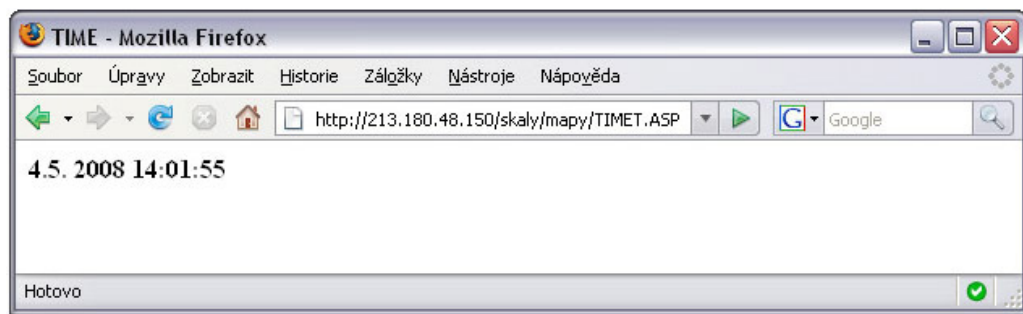
¹⁷ Velikost databáze do 4GB a z výpočetního výkonu serveru využije jen 1 CPU a 1 GB RAM.

¹⁸ Nejčastěji to jsou VB Script, JScript, Perl a Python. Jejich podpora je zajištěna instalací příslušných modulů.

Výhodou ASP je možnost využití široké škály skriptovacích jazyků, což otevírá dveře programátorům, kteří se nemusí zabývat učením dalšího jazyka. Nutností je pouze instalace příslušného jazykového překladače (*scripting engine*) na server. Ve všech později uváděných případech bude použit skriptovací jazyk VBScript. Na Obr.3.2 a Obr.3.3 je dobře zřetelný princip zpracování kódu na straně serveru a výsledek odeslaný klientskému prohlížeči.

<pre><html> <head> <title>TIME</title> </head> <body> <% @LANGUAGE=VBScript %> <% Response.Write(Now()) %> </body> </html></pre>	<pre><html> <head> <title>TIME</title> </head> <body> 4.5. 2008 13:50:57 </body> </html></pre>
---	---

Obr.3.2: Na pravé straně je zobrazen kód ASP před požadavkem klienta na jeho zobrazení. Pravá strana obsahuje kód, tak jak jej obdrží prohlížeč po jeho volání.



Obr.3.3: Takto se kód zobrazí v prohlížeči

Aby server poznal, že je po požadavku prohlížeče na zobrazení stránky nutné nejprve zpracovat kód ASP a později odeslat výsledek, adresa stránky nenese obvyklou příponu *.html*, ale právě koncovku *.asp*. To je také vidět v adresním řádku prohlížeče na Obr.3.3.

3.1.3 SQL

Pro přístup k informacím uložených v databázích (entitám, atributům) byl navrhnut jazyk syntakticky vycházející z anglického jazyka. Zahrnuje 4 skupiny nástrojů: definiční (DDL), řídicí (DCL), manipulační (DML), zobrazovací (VDL). Přičemž každá ze skupin je preferovaná jinou skupinou uživatelů. Například DDL

využijí správci databáze, kdežto VDL nástroje přísluší spíše skupině uživatelů na klientské úrovni. V našem případě bylo nutné pouze získat hodnoty z databáze a proto byl nejčastěji používán příkaz *SELECT* s parametry *FROM* a *WHERE*. Pro lepší představu je typický SQL dotaz používaný v aplikaci zobrazen na Obr.4.3.

3.2 Mapová komponenta

3.2.1 API amapy.cz založené na JavaScript a MooTools framework

Přestože může být JavaScript využit i na straně serveru jako server-side řešení, následující text i reálné užití pro mapovou aplikaci bylo napsáno pomocí rozšířenější client-side verze.

„JavaScript je objektově orientovaný skriptovací jazyk, syntakticky vycházející z rodiny jazyků C/C++/Java. Objektově orientované mechanismy se ale od této rodiny zcela liší. V kombinaci s DOM (viz. níže) dostáváme velmi silný nástroj pro rozhýbání obsahu webových stránek a tím i naplnění pojmu DHTML. JavaScript byl poprvé veřejně publikován 4. prosince 1995¹⁹ společnostmi Netscape a Sun. Jeho původní poslání, umožnit programátorům lepší kontrolu nad Java applety, vzalo rychle za své a našlo si vlastní cestu k životu v podobě manipulace s obrázky a s obsahem webových dokumentů.“²⁰

Mapové API využívá celou řadu moderních řešení založených na JavaScriptu. Dalo by se říci, že odpovídá definici poněkud sporného pojmu: web 2.0. Toto je výčet těch nejzásadnějších vylepšení, které stojí za zmínku:

- JSON²¹ – výměnný formát pro uchování a přístup k datovým strukturám. Mezi jeho přednosti patří podpora mnoha programovacích jazyků a oproti xml i menší datová náročnost.
- DOM – API umožňující přímý přístup k datovým strukturám HTML dokumentu. Je založen na stromové struktuře a díky tomu dokáže

¹⁹ Práce na něm však začala již zpočátku roku 95, kdy Brendan Eich stočil LiveScript, který byl až později a poněkud nešťastně přejmenován na JavaScript. Počáteční slabá podpora ze strany Internet Exploreru3 vedla Netscape a Sun ke standardizaci pomocí ECMA a v pozdější době i ISO.

²⁰ CHAMPEON, Steve. JavaScript: How Did We Get Here?. O'Reilly Network [online]. 2001 [cit. 2008-04-30]. Dostupný z WWW: <http://www.oreillynet.com/pub/a/javascript/2001/04/06/js_history.html>.

²¹ <http://www.json.org/>

rozlišovat hierarchii jednotlivých objektů. Pomocí JavaScript umí tyto objekty (elementy, atributy, text, atp.) snadno modifikovat. Rozlišujeme několik úrovní DOM (0-3), dle zahrnutých modulů pro práci s dokumenty.

- AJAX – technologie pro asynchronní výměnu dat. Tím je možné dosáhnout zobrazení určitých informací bez nutné obnovy (překreslení) celé stránky.

Výčet se neobejde bez poznámky, že jednotlivé technologie je velmi obtížné jednoznačně vymezit. Jejich největší síla tkví právě v jejich kooperaci. Také proto jsou základním pilířem knihovny MooTools²², která je v plném rozsahu k dispozici vývojářům mapových aplikací jako součást API, takže prakticky není třeba používat další externí framework knihovny. Pokud se přesto vývojář rozhodne využívat další externí knihovny, je třeba počítat s případnou kolizí jmen volaných metod. Jednou z možností, jak tomu předejít je využití tzv. *iframe*. Jeho popisem bychom se ale dostali nad rámec práce.

API dále nepřekvapí ani podporou vektorové grafiky, ať už v SVG anebo VML podobě. Jejich využití může sloužit pro zobrazení vektorových tras, případně pro vymezení hranic určitého území.

3.2.2 Souřadnicové systémy podporované API²³

Mapové API dokáže zpracovávat souřadnice ve 3 souřadnicových systémech: WGS84, S-JTSK a S-42 a je vybaveno implicitní funkcí, která je dokáže vzájemně převádět. Bohužel se mi nepodařilo zjistit, jaké algoritmy jsou pro tuto konverzi používány. Pro čistě informační účely by ale přesnost v řádech metrů neměla být na obtíž.

Formát vkládaných souřadnic může být různý. Jako oddělovač zeměpisné šířky a délky se může použít mezera, čárka, středník. Desetinný oddělovačem je vždy pouze tečka. Způsoby zápisu lze kombinovat.

Příklady podporovaných způsobů zápisu souřadnic:

²² <http://mootools.net/>

²³ <http://amapy.atlas.cz/api/principles.aspx>

- +15.347534
- +15.347534°
- 15.3434°N
- 15d30.344'N
- 15° 30.344'N
- 15° 30' 3.34"N

Obecně je ale možné v případě potřeby zavést podporu jiného souřadnicového systému naprogramováním vlastních funkcí.

3.2.3 HTML a CSS

Když na přelomu osmdesátých a devadesátých let výzkumníci CERNu navrhli jazyk HTML, určitě si stěží představili, jaký dopad to bude mít na celou společnost a jak se během pár let stane internet fenoménem doby. Samotné HTML²⁴ se v čisté podobě na internetu již téměř nevyskytuje. Na vině je jeho statičnost, která bývá často eliminována na straně uživatele i serveru využitím skriptovacích jazyků. Například jako je výše uvedený client-side JavaScript v kombinaci s kaskádovými styly – CSS a server-side PHP a ASP. Vlastní syntaxe zdrojového kódu se skládá z párových značek (tagů) a jejich atributů, které definují charakter kódu/textu mezi nimi, a také nepárových, které slouží například k vložení obrázku apod.

Pro práci s rozsáhlejšími dokumenty je dobré využít CSS. Tím se dosáhne oddělení formátování struktury obsahu webu a jeho vzhledu (stylu) a tím i lepší přehlednosti. Velkým kladem uložení těchto informací v samostatném souboru mimo vlastní HTML dokument je možnost využít ho jako společný pro více dokumentů a tím sjednotit výsledný design. Toto externí (statické) uložení platí jak pro soubory s příponou .css, tak i pro .js. V našem případě bylo CSS zahrnuto do těla webové stránky mapové aplikace.

²⁴ Mluvíme především o poslední verzi 4.01, která se deklaruje jako verze finální. Pokračovatel nese již zcela jiné označení XHTML 1 a přináší spíše omezení a komplikace s podporou ze strany prohlížečů, než zásadní přínos. Prakticky se jedná o reformulované HTML. Více informací na <http://www.w3.org>

4 REALIZACE MAPOVÉHO MODULU

Pro úpravu a přehlednost psaného zdrojového kódu byl použit freewarový nástroj PSPad. Mezi jeho přednosti patří podpora barevného rozlišení syntaxe a to nejen HTML a CSS, ale i ASP.

Výsledné ladění probíhalo na testovacím serveru, který využívá stejné technologie, jako server produkční. Jediná, ale o to větší technologická odlišnost testovacího serveru je v použitém kódování serveru, které bylo nastaveno jako win-1250. To bylo příčinou nekorektního zobrazení některých diakritických znaků, jejichž původ byl obsahem databáze tvořící popisnou složku mapy. Jak bylo uvedeno výše – databáze využívá kódování znakovou sadou UTF-8. Nebylo ale nutné řešit problém konverzí mezi znakovými sadami, protože tento stav neměl vliv na samotné ladění kódu. Navíc při následném nasazení na server veřejný bylo vše v pořádku a bez nutnosti zásahu.

Testovací server byl vybaven fragmentem databáze a byl zcela oddělen od databáze serveru produkčního. Výhoda velikosti testovací databáze byla v rychlé odezvě na časté požadavky, přístupy a obnovy laděných souborů mapové aplikace.

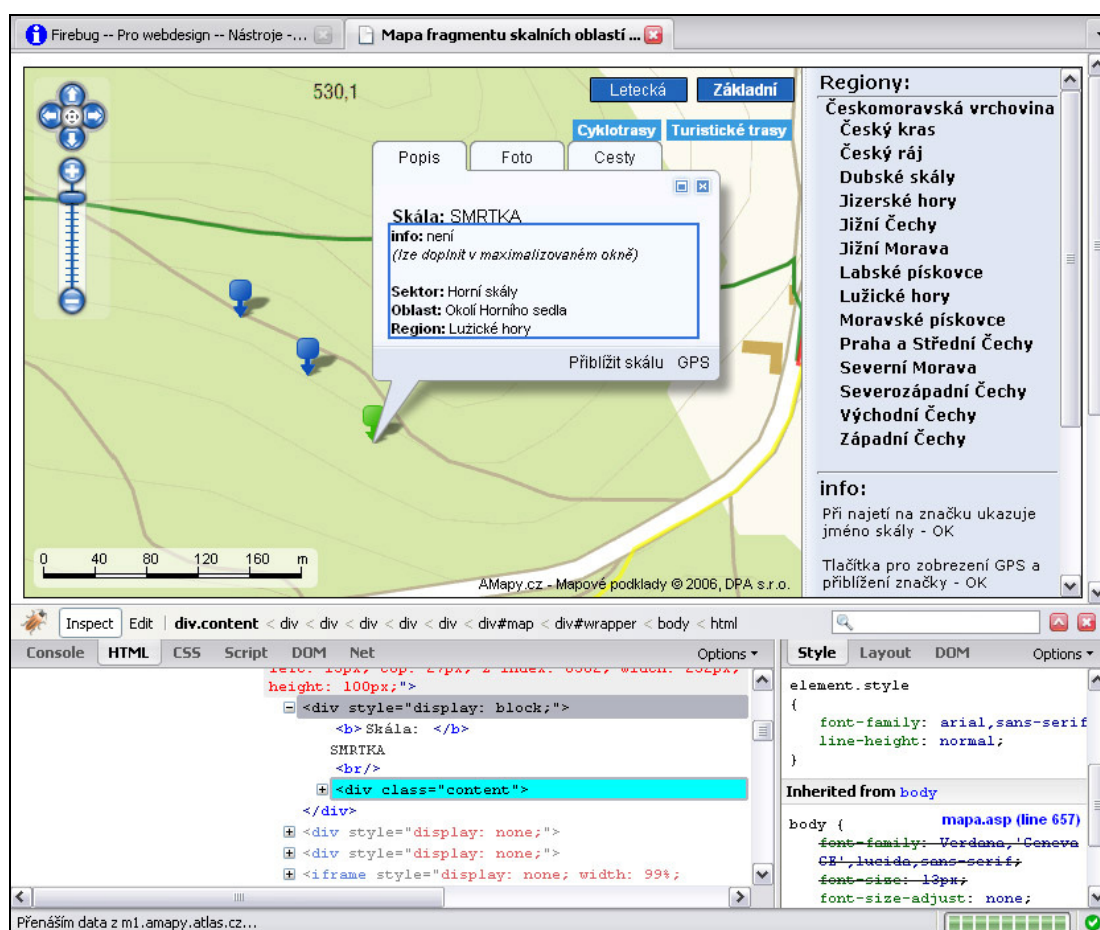
Pro samotné testování a ladění kódu byl použit výborný FireBug²⁵ 1.05 nainstalovaný jako doplněk prohlížeče Firefox 2.0.0.14. Ve spolupráci s ním prokázal výbornou službu a určitě si zaslouží krátké představení. Po vlastní instalaci se nám ve stavovém řádku zobrazí ikona symbolizující stav po načtení webové stránky a v případě

²⁵ <http://www.getfirebug.com/>

chyb i jejich počet. Samotné okno Firebugu se otevře po kliku na ikonu stavu a zobrazí se integrované v okně prohlížeče. Je možné jej ale oddělit do okna samostatného.

Následuje přehled základní nabídky včetně popisu funkcí:

- *Console* – základní informační okno s reportem případných chyb na stránce, včetně informací o spojeních mezi serverem a klientem metodami GET a POST, případně AJAX
- *HTML* – je možné procházet kód načtené stránky a to buď přímo, případně výběrem objektu myší přímo na webové stránce. Následně se zobrazí informace o CSS a *layout* prvku. Jednotlivé atributy a vlastnosti je možné přímo editovat (po stisku tlačítka *Edit*) a změna se v reálném čase projeví i na stránce. Velmi přehledně je řešeno i zobrazení dědičnosti stylů.
- *CSS* – zobrazí kompletní výpis kaskádových stylů aplikovaných na stránce
- *Script* – umožňuje pozorování hodnot proměnných a to i během běhu skriptu
- *DOM* – zobrazí kompletní výpis objektového modelu v abecedním pořadí
- *Net* – generuje seznam načítaných objektů, adresu požadavku, informaci o přenesených datových objemech a časových prodlevách.



Obr.4.1: Prostředí programu Firebug integrovaného do okna prohlížeče – prověření objektu²⁶

4.1 Principy tvorby mapy

Následující text je úzce spjat s Obr.4.2.

Jak bylo řečeno již v odstavci 2.5.5, pro možnost užití mapy na svých stránkách je nutná registrace domény, kde bude mapová aplikace provozována. Na základě této registrace uživatel obdrží e-mailem unikátní identifikační číslo, které je součástí odkazu na JavaScript, který musí být uveden v hlavičce (*head*) HTML dokumentu. Samotné načtení mapy je lehce komplikováno faktem, že načtení mapy událostí *load* může být neúspěšné v případě, že se nepodaří načíst veškerá data tvořící mapový obsah (nestažený obrázek, atp.). Mapové API vrací událost *domready*²⁷ proměnné *Page*. Tato událost má za následek zobrazení mapy bez závislosti na úspěšném načtení všech

²⁶ V pravém dolním rohu je situován symbol stavu načtení stránky. V tomto případě Firebug neidentifikoval žádnou chybu kódu.

²⁷ Řešení přinesla trojice vývojářů Dean Edwards, Matthias Miller a John Resig. Jejich fix upravuje problém s užitím handleru události *window.onload*. Ten totiž čeká, až se např. obrázky, soubory načtou a znemožní vykonání dalšího kódu.

obrázků tvořících mapový podklad, případně objektů tvořící náplň informačních bublin apod. Dále je nezbytně nutné uvnitř proměnné *Page* deklarovat proměnnou typu *AMap* (v našem případě se jmenuje *mainMap*) a následně na ni aplikovat metodu *loadMaps*. Ta může obsahovat následující argumenty:

- *center* – definuje střed mapy,
- *scaleOrScaleIdx* – definuje měřítko,
- *mapTypeName* – nastaví jména pro přepínače typů map,
- *mouseCenter* – pro určení, zda-li se střed mapy nastaví dle aktuální polohy kurzoru myši. Mapa se vykreslí do *div* elementu ve kterém se zároveň definují i její rozměry.

```
<html>

  <head>
    <title>MAPA</title>
    <script type="text/javascript"
      src="http://amapy.atlas.cz/api/api.ashx?guid=zde bude id">
    </script>
    <script type="text/javascript">

      var Page = {
        load: function() {
          var mainMap = new AMap("mainMap");
          mainMap.loadMaps();
        }
      }

      window.addEvent('domready', Page.load.bind(Page));
    </script>
  </head>

  <body>
    <div id="mainMap" style="width: 800px; height: 600px;">
    </div>
  </body>

</html>
```

Obr.4.2: Ukázka minimálního kódu nutného pro vykreslení mapy

Tento jednoduchý kód má za následek vykreslení mapy bez ovládacích prvků, bez přepínačů vrstev, bez POI. Ale je již možné ovládat pomocí myši (posun, přiblížení kolečkem). V případě, že má metoda *loadMaps* nulové argumenty, nastaví se implicitní

střed a měřítko mapy tak, že je zobrazena Česká republika. V následujících odstavcích se postupně zaměříme na další prvky mapy.

4.2 Sestavení *Recordset*

Pro práci s informacemi z databáze je nejprve nutné definovat proměnné, které příslušné informace obsahují. Těmi jsou instance objektu *recordset*, které obsahují množinu výsledků vrácených po předchozím SQL dotazu databáze. Tyto SQL dotazy jsou sestavovány s ohledem na cíl užití a to tak, že výsledek zahrnuje jen entity a atributy, které jsou potřeba k dalšímu zpracování. Výsledky dotazů se jazykem ASP předávají proměnným, které tvoří obsah bublin, informačního panelu, atp. Tabulky, které byly použity pro tvorbu *recordset* záznamů jsou viditelné na diagramu Obr.3.1. Přehledný souhrn všech *recordset* záznamů, jejich proměnných, zdrojových tabulek a podmínek je znázorněn v Tab. 4.1.

RecordSet	SELECT	FROM	WHERE
rsSKALA	sektor_id nazev gps id	skaly_skala	
rsLIST	id nazev gps	skaly_skala	
rsSEKTOR	oblast_id nazev	skaly_sektor	id=" & rsSKALA("sektor_id") & "
rsOBLAST	region_id nazev	skaly_oblast	id=" & rsSEKTOR("oblast_id") & "
rsREGION	nazev	skaly_region	id=" & rsOBLAST("region_id") & "
rsSKALAFOTO	foto	skaly_skala_foto	skala_id=" & rsSKALA("id") & "
rsREG	id nazev	skaly_region	

Tab. 4.1: Přehled *recordset* záznamů použitých pro tvorbu mapového obsahu

Jeden z problémů, který se projevil při zpracování a načítání proměnných do funkcí tvořících mapu, byl formát zápisu souřadnic skal v databázi. Souřadnice pro zeměpisnou délku i šířku jsou v souřadnicovém systému WGS84 a jsou psané jako jeden *string* např. takto: 48°50'32.98"N,14°22'27.64"E. Vzhledem k tomu, že znak vteřin (") je interpretován jazykem JavaScript, bylo nutné ho získat ve tvaru, který tyto znaky neobsahuje (konverze na desetinný tvar), a nebo ve tvaru, který využívá speciálního zápisu: 48°50'32.98\\'N,14°22'27.64\\'E. Obrácená lomítka před uvozovkami totiž slouží jako takzvaná *escape* sekvence, která není jazykem JavaScript interpretována²⁸. Doplnění těchto předřadných znaků se provede přímo v sestavení SQL dotazu pomocí jednoduchého příkazu *replace*, jak je vidno na Obr.4.3.

²⁸ V jazyce JavaScript se totiž symboly uvozovek používají pro ohraničení textových hodnot proměnných.

```
set rsSKALA = con.execute("SELECT id , sektor_id, nazev,  
replace(gps, '"', '\"') AS gps, poznamka FROM skaly_skala")
```

Obr.4.3: Sestavení SQL dotazu s využitím příkazu *replace*

4.3 Navigační a kompoziční prvky

Součástí každé interaktivní mapy jsou nutně ovládací prvky. Jejich kvalita, přehlednost a jednoduchost jsou jedny z prvotních předpokladů pro vytvoření příjemného uživatelského prostředí, ke kterému se bude uživatel rád vracet. Implicitní řešení nabízené API tyto předpoklady splňuje bez výhrad a pro specifitější požadavky nabízí celou řadu nástrojů a funkcí, které umožňují tvorbu navigačních i kompozičních prvků na míru konkrétnímu webu. Jedna z motivací tvorby vlastních prvků by mohla být otázka designu a korespondence vzhledu mapy s okolním obsahem webové stránky.

4.3.1 Navigační panel

Mapové API je vybaveno dvěma již předdefinovanými ovládacími prvky. Oba jsou potomci třídy *AMapPartBase* a jak je pro tuto třídu zvykem, je možné si ovládání vytvořit či přizpůsobit zcela dle své představy. Pro účely této práce nebylo nutné ani účelné navigační prvky modifikovat a bylo použito komplexnější z obou řešení. Tím je třída *AMapControl*, která obsahuje tlačítka posunu mapy a zároveň i posuvník ovládající mapové měřítko (viditelný v levém horním rohu Obr.4.1). Inicializace panelu je velmi jednoduchá: třída *AMapControl* se předá potomkovi třídy *AMap* metodou *addMapPart*.

```
this.mainMap.addMapPart(new AMapControl());
```

Obr.4.4: Přidání navigačního panelu do mapy

4.3.2 Grafické měřítko

Problematika tvorby grafického měřítka není tak jednoduchá, jak by se mohlo na první pohled zdát. API disponuje možností grafické měřítka zobrazit. Jedná se ale o nezdokumentovanou funkcionalitu vytvořenou pravděpodobně těsně před přerušáním vývoje. Třída se jmenuje *AScaleOfMap* a jedná se jako obvykle o potomka třídy *AMapPartBase*. Do mapy se přidá pomocí metody *addMapPart*. Mapové měřítko se vykresluje vždy tak, aby hodnota již vymezuje byla vždy celé číslo. Tento fakt má za

následek, že se jeho velikost při změně měřítka mapy mění. S největší pravděpodobností jsou jeho velikosti a hodnoty předdefinované a změnou měřítka se načte i příslušný grafický etalon včetně popisu. Zajímavé je, že hodnota měřítka, která se zobrazí při najetí na posuvník navigačního panelu²⁹ je uváděna jako poloviční, než by měla být. Nevím, z jakého důvodu API používá tyto nepravdivé hodnoty, ale je jisté, že by je nemělo používat veřejně. Jednoznačně to vede k desinformaci uživatelů v případě, že není zobrazeno měřítko grafické. Chtěl jsem proto vytvořit poněkud komplexnější a přehlednější grafický prvek s více dílky a při zkoumání možností této problematiky jsem narazil na několik problémů. Mnou navrhnutá grafická podoba má vzdálenost dílců $k = 40 \text{ px}$. Pro každý druhý dílec se při změně přiblížení mapy spočítá odpovídající hodnota v kilometrech, případně metrech. Funkce třídy `AMap` `getCurrentScaleInfo().coefficientX` vrátí poměr mezi šířkou dílce v metrech a jeho šířkou v pixelech. Tato hodnota se vynásobí koeficientem k a pro vzdálenost v kilometrech je nutné ji dělit tisícem. Celý tento proces je naprosto exaktní a to je také jeho jedinou nevýhodou. Hodnoty dílců totiž nejsou celá čísla a to má za následek „neučesanost“ výsledku jak je vidět na Obr.4.5. Proto tedy mapová aplikace obsahuje původní, leč nezdokumentovanou podobu grafického měřítka.



Obr.4.5: Grafické měřítko

4.3.3 Přepínače vrstev

Součástí mapového okna jsou tlačítka sloužící k obsluze typů zobrazovaných map (*Letecká, Základní*)³⁰ a doplňkových překryvných vrstev (*Cyklotrasy, Turistické trasy*). Jejich ovládací prvky se připojí velice jednoduše. Mapy i vrstvy jsou potomci třídy `MapPartBase` a do mapy se přidají obvyklou metodou `addMapPart` jak je vidět na Obr.4.6. Překryvné vrstvy turistických tras a cyklotras jsou viditelné od měřítka 1 : 500 000.

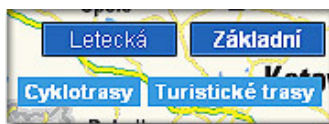
²⁹ Stejně hodnotu vrací i funkce `getCurrentScale()`.

³⁰ Jejich pojmenování uvedené na tlačítkách je možné změnit v argumentu metody `loadMaps` uvedené v odst. 4.1.

```
this.mainMap.addMapPart(new AMapTypeControl());  
this.mainMap.addMapPart(new AMapLayerControl(  
    [A_CYCLE_MAP, A_TOURISTIC_MAP]));
```

Obr.4.6: Přidání mapových prvků spravující mapové vrstvy

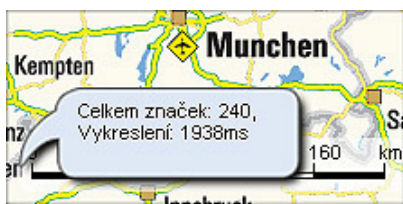
Pokud bychom chtěli zobrazit mapu základní a přes ni s určitým stupněm průhlednosti mapu leteckou, existují nástroje, jak tohoto efektu snadno dosáhnout. Výsledek sice vypadá „efektně“, ale určitě si nenajde širší uplatnění a práci s mapou činí spíše nepřehlednou.



Obr.4.7: Přepínače typů map a překryvných vrstev

4.3.4 Informační bublina

Jak již název sám napovídá, tato bublina má za úkol informovat o počtu načtených POI a také o čase, za který dojde k jejich zobrazení. Informační bublina se implicitně zobrazí v levém dolním rohu mapy po načtení celé stránky (viz. Obr.4.8). Doba zobrazení byla nastavena na 3000 ms. Po uplynutí tohoto intervalu bublina zmizí.



Obr.4.8: Informační bublina

4.4 POI a elastická bublina

Prvek mapy elastická bublina slouží k zobrazení podrobných údajů o skále. Ta se zobrazí vždy po stisku levého tlačítka myši na ikonu skály (POI). S ohledem na povahu zobrazovaných informací jsem navrhl rozdělení bubliny na tři záložky, které mají společnou patičku. V pravém horním rohu se nalézá tlačítko maximalizace pro zobrazení podrobného seznamu horolezeckých výstupů (cest). To je řešeno načtením HTML stránky do maximalizovaného okna. Dále bublina obsahuje také klasický křížek pro zavření okna, tak jak jej známe z platformy MS Windows. Na mapě může být vždy

otevřená pouze jedna bublina. Při otevření bublině a požadavku na zobrazení další se původní zavře a zobrazí bublina požadovaného POI.

```
var marker = new AMarker(gps, {
    title      : title ,
    label      : "" ,
    draggable  : true,
    id         : id
});
marker.addEvent('onClick', function() {
    marker.showBubble(tabs, {
        footerContent : footerContent,
        'maxSize'     : false,
        'Width'       : '300',
        'Height'      : '400',
        'maxUrl'      : 'mapa_combo.asp?skala_id=' + id + ' '
    });
});
```

Obr.4.9: Deklarace proměnné *marker* typu *AMarker*

Hlavní náplní bubliny jsou data z databáze Skalních oblastí. Ta jsou jako obvykle načítána cyklem v ASP skriptu, který je součástí hlavního mapového souboru³¹. Tento skript má za úkol procházet entity a atributy konkrétních *recordset* záznamů a ukládat je do příslušných proměnných JavaScriptu. Na konci každého průběhu cyklu je volána funkce *createMarker()*. Ta má za úkol předat proměnné příslušným třídám API, vytvořit značku na mapě a těmto značkám přidat událost (*marker.addEvent*, viz Obr.4.9) zobrazení bubliny po kliku myší. Na konci funkce *createMarker* se značka s propojenou událostí přidá do pole *markers*. Proměnné z tohoto pole následně metodě *addOverlay* třídy *AMap*, která POI vykreslí do mapy.

4.4.1 Záložky bubliny

Využití záložek přináší řád do bubliny. Každá z nich má specifické určení, které je tématicky nesourodé. Nastavení záložek se řídí pomocí *options* bubliny. Přebírá její výšku a šířku, ale pokud se text do záložky nevejde, automaticky se aktivuje posuvník po pravé straně okna.

- záložka *Popis* – (ve zdrojovém kódu značená *tab1*) dle mého názoru je nutné zde zobrazit ty nejnutnější informace, protože jsou viditelné ihned

³¹ Soubor *mapa.asp* v kořenovém adresáři na CD.

po otevření bubliny. Proto tato položka obsahuje název skály, sektoru, oblasti a regionu.

- záložka *Foto* – (ve zdrojovém kódu značená *tab2*) obsahuje galerii fotografií, pokud je obsahem databáze. V případě že není, zobrazí se jednotný obrázek informující o nedostupnosti fotografie. Výhodou záložky foto je možnost použít integrovanou obrázkovou galerii. Funkčně se jedná o lehce modifikovaný klon *LightBox*³². Menší nedostatek je zobrazení náhledů zmenšením obrázků skutečné velikosti. To má za následek poněkud větší objem přenesených dat. Velikost náhledů je lehce nastavitelná pomocí parametru *imageThumbnailHeight* (viz Obr.4.10).
- záložka *Cesty* – (ve zdrojovém kódu značená *tab3*) obsahuje upozornění, na zobrazení informací o skalních cestách po stisku tlačítka maximalizace. Do budoucna je možné doplnit její obsah například o statistiku počtu cest v konkrétní obtížnosti, případně zprávou o zákazu provozování horolezecké činnosti z důvodu hnízdění chráněného ptactva apod.

4.4.2 Patička bubliny a její funkce

Tento element (*footerContent*) se volá v *options* bubliny. Je společný pro všechny záložky a je viditelný i při maximalizovaném okně. Obsahuje dvě tlačítka: *Přiblížit skálu* a *GPS*. Obsahem patky může být DOM element, HTML string případně objekt typu *ABubbleFooterTab*.

```
var tab1 = new ABubbleTab("Popis", popis);
var tab2 = new ABubbleTab("Foto", images,
    {'imageThumbnailHeight': '70'});
var tab3 = new ABubbleTab("Cesty", "Stiskněte tlačítko maximalizace
    pro více informací!<br />");
var tabs = [tab1, tab2, tab3];

var footerContent = [
    new ABubbleFooterTab( 'GPS', this.showGps.bind(this)),
    new ABubbleFooterTab('Přiblížit skálu', this.Zoom.bind(this))
];
```

Obr.4.10: Definice záložek a patičky bubliny

³² <http://www.huddletogether.com/projects/lightbox2/>

Stisk tlačítka *Přiblížit skálu* volá funkci *zoom()*. Ta má za úkol načíst souřadnice aktuálně otevřené bubliny, ty předat mapě včetně parametru přiblížení. Mapová funkce třídy *AMap zoomTo()* tyto parametry přebere a výsledek interpretuje nastavením středu mapového okna dle GPS souřadnic bubliny a nastaví stupeň přiblížení. Stupeň přiblížení je nastaven na měřítko 1:4000, přestože by bylo možné zobrazit mapu i v měřítku 1:2000. Pro letecké snímky je měřítko limitováno hodnotou 1:3000. Pokud by se tedy POI zobrazil v maximálním možném stupni zvětšení pro základní mapy, pro letecké snímky by podklad nebyl nalezen.

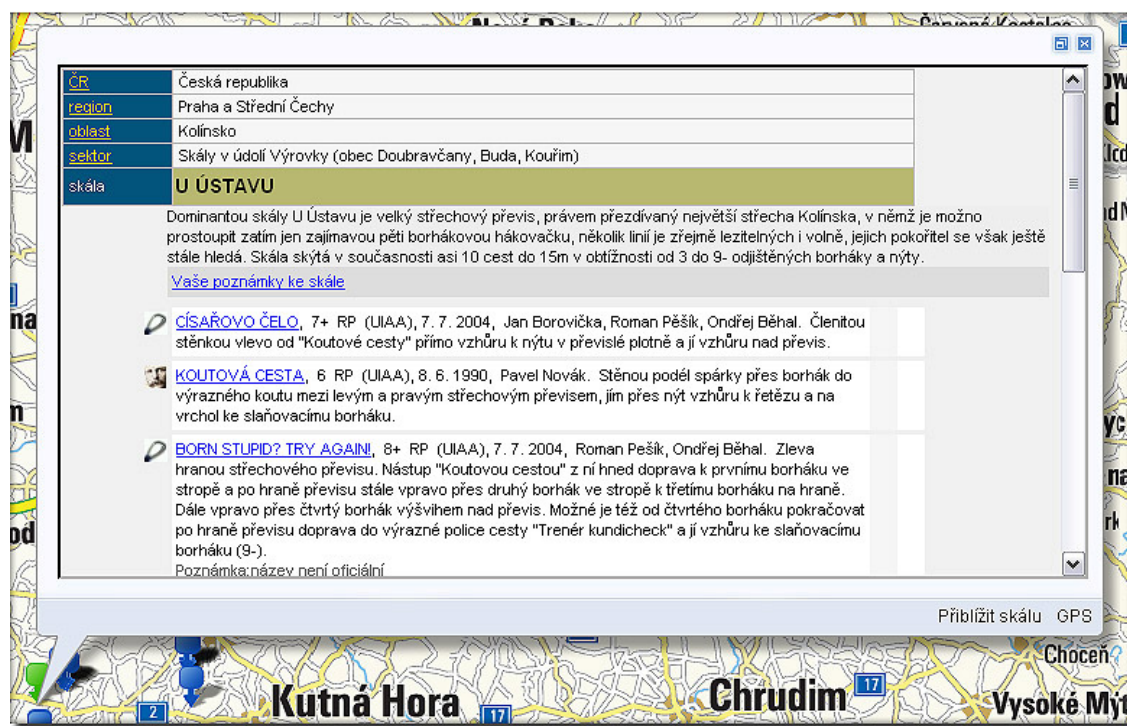
Lehkou úpravou kódu by se dal mapový podklad po přiblížení automaticky přepnout na podklad leteckého snímku, případně na vrstvy cyklotras apod.

Tlačítko GPS využívá stejného principu použitým ve funkci *zoom()*. Zjistí souřadnice POI otevřené bubliny a ty po kliku zobrazí v patičce bubliny. Ve tvaru stupňů minut a vteřin³³.

4.4.3 Okno maximalizace

Jak bylo řečeno hned v úvodu, maximalizace bubliny je řešena pomocí URL. V cílovém zvětšeném okně se zobrazí požadovaná stránka. Protože máme uloženou proměnnou *id* skály, dá se tato hodnota použít pro sestavení URL (viz. Obr.4.9). Zadávaná URL odkazuje na stránku stávající podoby webu. V maximalizovaném okně je dále možné zobrazit podrobný výpis konkrétní cesty, případně výstup zapsat do virtuální vrcholové knihy. V hlavičce zobrazeného výpisu skály je přehledný model struktury oblastí, regionů a sektorů. Fungují jako odkaz na zobrazení obecnějších informací o daném místě. Dále je zde formulář pro přidání komentáře k právě otevřené skále (*Vaše poznámky ke skále*).

³³ Tento formát byl zvolen, protože se jeví přehlednější než desetinná varianta. API obsahuje nástroj pro přepočet souřadnic do systému S-JTSK a dále i S-42.



Obr.4.11: Maximalizovaná bublina s podrobným výpisem skály

Při testování kódu byla objevena drobná chyba API. V případě že je bublina maximalizovaná a opět se zmenší (normalizuje), API nenastaví původní hodnoty rozměru bubliny tak, jak jsou nastavené v jejím *options*, ale nastaví hodnoty implicitní. I přes snahu o nápravu této chyby a kontaktování vývojářů se náprava nerealizovala³⁴. Podobná chyba se projevuje i ihned po načtení mapové aplikace. Vnitřní logika API totiž nebere pevně nastavené hodnoty šířky a výšky bubliny závazně, ale bublinu jako takovou zmenší v případě, že text libovolné ze záložek ji zcela nenaplní. Tento efekt jde proti logice věci.

„Bublina má sice svoji výchozí šířku a maximální výšku, ale tyto hodnoty lze upravit. Maximální výška bubliny znamená, že pokud obsah přeteče, objeví se posuvník a naopak, je-li obsah menší, bublina se mu výškou přizpůsobí.“³⁵

³⁴ Vývoj API se prakticky zastavil fúzí společností *Atlas.cz* a *Centrum.cz* (více na <http://www.spojeno.cz>). Tato fúze měla zásadní vliv na odchod vývojářů API v čele s Danielem Steigerwaldem z firmy *Atlas.cz*.

³⁵ Průvodce po API [online]. [2007] [cit. 2008-04-20]. Dostupný z WWW: <http://amapy.atlas.cz/api/examples.aspx#f>.

4.5 Informační panel – *Sidebar*

Tento prvek, stejně jako vyhledávací okno, si klade za úkol pomoci uživateli v rychlé orientaci a navigaci za kýženou informací. Obsahuje kompletní výpis všech regionů obsažených v databázi. Při výběru regionu se následně rozbalí menu a je zobrazen souhrnný seznam jeho oblastí. Oblasti následně slouží jako odkazy do mapy. Při výběru požadované oblasti funkce předá mapě potřebné parametry přiblížení a polohy. V dolní části panelu je oddíl *Poznámky* určený pro text informativního charakteru vztahující se k ovládání mapové aplikace a případné umístění legendy mapových objektů.

Panel je situován na pravé straně mapového okna a má tu vlastnost, že po provedení stisku levého tlačítka myši nad jeho levým okrajem se schová. To má za následek zvětšení viditelné části mapy. V případě požadavku uživatele na opětovné zobrazení panelu stačí vykonat stejnou akci na jeho stále viditelný okraj.

```
<div id="listContainer">
  <h1>Regiony:</h1>

  <% set rsREGION = con.execute("SELECT id, nazev FROM
    skaly_region ORDER BY UPPER(nazev)") %>
  <% if not rsREGION.eof then %>
  <% do while not rsREGION.eof %>

    <div class='placeitem'>
      <h2 onclick="showHide('reg<%= rsREGION("id") %>') ">
        <%= rsREGION("nazev") %></h2>
      <div id="reg<%= rsREGION("id") %>" class="hide"></div>
    </div>

  <% rsREGION.movenext %>
  <% if err.number<>0 then exit do %>
  <% loop %>
  <% end if %>

</div>
```

Obr.4.12: ASP skript pro tvorbu seznamu regionů v panelu *Sidebar*

4.5.1 Obsah panelu

Jména regionů jsou odstavce třídy *placeitem* předka *listContainer*, které jsou vygenerovány pomocí ASP skriptu. Každý region má svého potomka v podobě odstavce, který má unikátní *id* (viz Obr.4.12). Jak je vidno na stejném obrázku, každý element *h2* registruje událost po stisku myši a ta volá funkci *showHide()* jejímž úkolem je právě zobrazení/schování elementu *div*, který obsahuje oblasti toho kterého regionu.

Díky této metodě schovávání/rozkrývání je možné udržovat menu přehledné a příjemné ho vřít.

Při tvorbě panelu jsem narazil na problém, jak mapě po kliku na oblast předat nejen polohu, ale i úroveň přiblížení. Mezi jednotlivými oblastmi jsou totiž značné rozdíly v ploše území, které zastupují. Jedna z teoretických možností byla zobrazit všechny skály na mapě, které do daného území (oblasti) patří³⁶. Tento způsob je ale prakticky nerealizovatelný. Překážka tkví v samotné databázi. Nutným předpokladem zobrazení skály je informace o její poloze. Řada skal v databázi ale tuto informaci, alespoň prozatím, nemá. Samotná databáze je neustále doplňována a záleží na správci oblastí³⁷, aby GPS souřadnice řádně zapsali. Tento systém funguje poměrně úspěšně, ale zatím se jedná pouze o několik oblastí, které jsou kompletní. Je tedy jasné, že zobrazení území by v tomto případě neprobíhalo vždy korektně. Proto bylo nutné vymyslet jiný způsob, který není založen na automatizovaných postupech a exaktních funkcích, ale naopak si vyžádal trochu mechanické práce. Pro každý region jsem vytvořil pole obsahující seznam všech jeho oblastí³⁸ a každé oblasti přiřadil GPS souřadnice středu mapy a informace o měřítku tak, aby co nejlépe odpovídaly danému území. Název pole se skládá z neměnného *reg* a *id* odpovídající příslušnému *id* regionu z databáze.

³⁶ To by mohlo být založené na určení extrémních hodnot souřadnic všech skal spadajících do dané oblasti a na jejich základě vypočítat střed mapy a měřítko vhodné pro jejich zobrazení.

³⁷ Mapový server by měl správce motivovat, protože zobrazená skála/oblast na mapě slouží i k propagaci lezeckých oblastí mezi komunitou lezců.

³⁸ V době zpracování projektu databáze obsahovala 110 oblastí.



Obr.4.13: Informační panel - výběr oblasti

4.5.2 Funkce panelu

Spojení položky v menu s funkcí změny polohy a měřítka mapy zajišťuje funkce *createJumps()*. Jak je vidět na Obr.4.14, funkci se předá proměnná *places*, což jsou pole s regiony. Funkce vytvoří element *div* třídy *item*. Tento *div* je potomek odstavce tvořícího jednotlivé regiony. Neboli `<div class='item'>` je obsah, který funkce *showHide()* zobrazuje/skrývá (viz. odstavec 4.5.1)

Při rozbalení regionu a najetí myši na oblast dojde k barevnému zvýraznění této položky. To je zajištěno funkcí *onItemOverOut()*. Při výběru položky dojde ke spuštění funkce *onItemClick()*, ta předá mapě nový střed a stupeň přiblížení odpovídající vybrané oblasti. Obě tyto funkce jsou vykonány na základě událostí *mouseover*, *mouseout* a *click*, které jsou přidány odstavci *listContainer*. Ten sice obsahuje i odstavce s regiony, ale k výkonu obou funkcí dojde jen v případě, že odstavec má definovanou třídu *item*.

```
createJumps: function(places, listContainer) {  
    for(var i=0, l=places.length; i<l; i++) {  
        var mark = new AMarker(new AGeoPoint(places[i].value), {});  
        var item = new Element('div').addClass('item').setHTML(  
            places[i].id).disableSelection().injectInside(listContainer);  
        item.$tmp.zoom = places[i].zoom;  
        item.$tmp.marker = mark;  
    }  
}
```

Obr.4.14: Funkce *createJumps()* - tvoří položky menu a proměnné s parametry polohy a přiblížení

4.6 Kontextové menu a jeho funkce

Stiskem pravého tlačítka uvnitř mapového okna se zavolá kontextové menu. Jeho okno obsahuje celkem tři položky:

- *Vycentrovat* – nastaví střed mapy dle místa, kde bylo stisknuto pravé tlačítko myši
- *Vyhledat skálu* – aktivuje vyhledávací panel

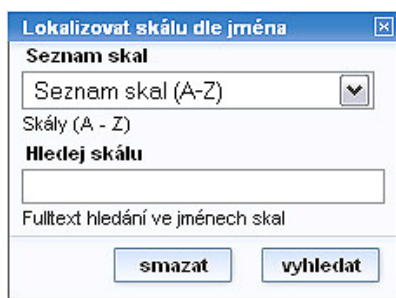
Poslední položka v seznamu nevyvolá další akci, ale zobrazí souřadnice systému WGS84 odpovídající místu stisku tlačítka myši. Jsou uvedeny v obvyklém tvaru stupňů, minut a vteřin.



Obr.4.15: Kontextové menu – poblíž menu je viditelný kurzor, ke kterému jsou vztaženy zobrazené souřadnice

4.7 Vyhledávací panel

Pro snadnou orientaci v mapě bylo nutné implementovat vyhledávání. To si klade za cíl vylepšit přívětivost webu a výrazně napomoci k urychlení navigace. Pro tuto funkci jsem navrhnul plovoucí panel, založený na třídě obsažené v API. Tuto třídu bylo nutné rozšířit a upravit pro naší potřebu. Tento panel je přístupný po výběru položky *Vyhledat skálu dle jména..* z kontextového menu (viz výše v této kapitole).



Obr.4.16: Okno pro vyhledávání skal

K vytváření formulářů v plovoucích oknech slouží třída *AMapPartDraggable*, která perfektně splňuje naše požadavky. Zavoláním potomka této třídy získáme předdefinovaný objekt mapy nacházející se nad všemi ostatními prvky. Tato třída je sama o sobě *div* element, jehož obsahem jsou v našem případě formulářové pole typu *Select* obsahující abecední seznam všech skal na mapě, dále pole typu *Input* pro fulltextové vyhledávání a konečně stavový řádek s tlačítky *smazat* a *vyhledat*. Výhodou této třídy jsou předdefinované kaskádové styly, které aplikují kompaktní vzhled na jednotlivé elementy a dále možnost interakce s mapou díky volání metody s parametrem *map*.

4.7.1 Nabídka – formulář typu *Select*

Protože se dá předpokládat rozšiřování databáze a samozřejmě i doplňování informací o poloze skal již v databázi obsažených, bylo nutné tomuto faktu přizpůsobit i obsah nabídky *Select*. Použité řešení je založené na principu cyklu v ASP, který prochází databází skal a hledá právě ty skály, u kterých je uvedena informace o prostorovém určení. Cyklus nám vygeneruje *option* a do jeho hodnoty *value* přidá *id* konkrétní skály.

```

"<select
onchange='javascript:zoomChoice(this.options[this.selectedIndex].value);' class='field' name='azlist' id='azlist'>" +
"<option selected value='default'>Seznam skal (A-Z)</option>" +
"<option>-----</option>" +

// Vytvoření seznamu všech skal - seřazeno a-z
  <% if not rsLIST.eof then %>
    <% do while not rsLIST.eof %>
      <% if rsLIST("gps") > "" then %>

        "<option value='<%= rsLIST("id") %>'><%= rsLIST("nazev") %>
        </option>;" +

      <% end if %>
      <% rsLIST.movenext %>
      <% if err.number<>0 then exit do %>
      <% loop %>
      <% end if %>

"</select>" +

```

Obr.4.17: Generování formuláře typu Select pomocí ASP skriptu

Jak je vidět na Obr.4.17, při výběru položky z roletového menu se spustí globální funkce *zoomChoice()* (viz Obr.4.18), která porovnává hodnotu *id* obsaženou ve formuláři typu *select* s příslušnými hodnotami v poli *markers* a při nalezení stejné hodnoty nastaví odpovídající hodnotu GPS dané skály jako střed a změní měřítko mapy na hodnotu 1:4000.

```

function zoomChoice(opt){
  if (opt != ''){
    for (i = 0; i < markers.length; i++) {
      if (markers[i].options.id == opt) {
        var gps = markers[i].getGeoPoint();
        this.mainMap.zoomTo(8000, gps, false);
      }
    }
  }
};

```

Obr.4.18: Funkce *zoomChoice()*³⁹

4.7.2 Vyhledávání – formulářové pole typu *Input*

Fulltextové vyhledávání funguje na principu porovnání zadané fráze/hesla s obsahem databáze. Tento způsob má ale jednu zásadní nevýhodu: hledaná fráze/heslo

³⁹ Hodnota 8000 předávaná mapě je ve skutečnosti dvojnásobná, než reálné měřítko.

musí být napsáno bez překlepů. Chtěl jsem proto využít technologie AJAX pro tvorbu tzv. *našeptávače*⁴⁰. Ten je schopen komunikací s databázovým serverem během psaní vyhledávaného slova v reálném čase sestavovat seznam, který zahrnuje slova odpovídající momentálnímu požadavku. Seznam se objeví v plovoucím okně pod *Input boxem* a kurzorem je možné vybrat námi hledané slovo. Evidentní je značné urychlení práce.

Během realizace se ale projevila značná časová náročnost a komplikovanost samotného řešení, a proto byl nakonec použit nástroj *AutoSuggest*⁴¹. Ten je k dispozici bez licenčních omezení⁴² a nabízí komplexní balík funkcí, včetně přívětivého designu okna výsledů, které je možné upravit jednoduchým zásahem do nastavení CSS.

Samotná jeho implementace do webové stránky spočívá v nastavení cesty ke skriptovému souboru našeptávače a zavolání příslušné třídy *AutoSuggest*. Dále bylo nutné vytvořit ASP skript⁴³, kterému pomocí AJAXové metody *GET Autosuggest* předá parametr s vyhledávacím *stringem*. Skript na jeho žádost sestaví odpověď a tu odešle zpět. Tato odpověď je sestavena ve formátu JSON (viz. kapitola 3.2.1). *AutoSuggest* na základě informace o *id* textového pole automaticky vytvoří plovoucí okno, které se objeví pod příslušným textovým polem. Součástí informací v našeptávacím okně je jméno skály a pro upřesnění i její sektor/oblast (viz Obr. 2.1). Skript souboru *find.asp* používá *SQL* dotaz, který porovnává zadanou hodnotu na vstupu (proměnná *q*) se jménem skály (*název*). Přesněji: ověřuje, zda-li *string* na vstupu není obsažen ve jménu skály (viz Obr.4.19).

```
sql = "SELECT id, sektor_id, nazev, gps FROM skaly_skala  
WHERE nazev LIKE '%" + q + "%' ORDER BY upper(nazev) "
```

Obr.4.19: Sestavení SQL dotazu v souboru *find.asp*

V nabídce je možné se pohybovat buď myší anebo šipkami klávesnice. Potvrzení volby z okna našeptávače se realizuje typicky klikem levé myši a stiskem klávesy *enter*. Proces vyhledávání se aktivuje stiskem tlačítka *vyhledat*.

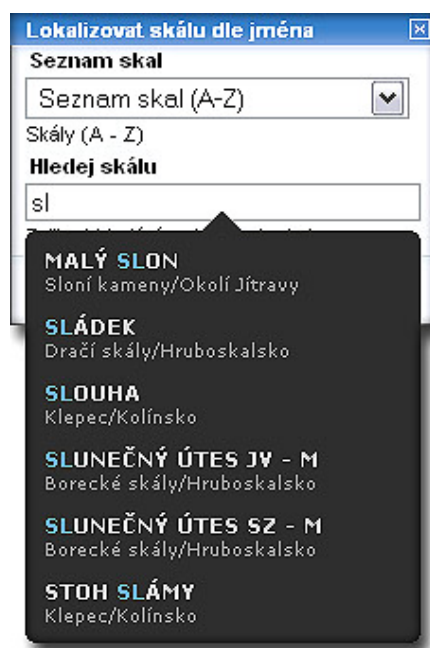
⁴⁰ Našeptávač je okno s výsledky vyhledávání, případně pravděpodobných hesel, které se zobrazuje pod textovým polem typu *input*, kde je možné výběrem položky urychlit proces vyhledávání. S nápadem využít našeptávač se můžeme setkat například na oblíbeném vyhledávací www.google.com

⁴¹ http://www.brandspankingnew.net/archive/2006/08/ajax_auto-suggest_auto-complete.html

⁴² <http://creativecommons.org/licenses/by/3.0/>

⁴³ Soubor *find.asp* obsažený v kořenovém adresáři CD přílohy

Okno vyhledávání je možné dále rozšířit o propojené *select* formuláře. Při výběru regionu by se s využitím AJAXu načetly jeho oblasti do dalšího roletového menu. Dalším upřesněním by se stejnou metodou návazně dalo dojít až na úroveň skal a dokonce i horolezeckých výstupových směrů. Při výběru konkrétní cesty by se mohl její výpis zobrazit v novém informačním okně.



Obr.4.20: Okno vyhledávání s "našeptávačem"

Dále by se vyhledávání dalo rozšířit i na jména měst, obcí, názvy geomorfologických prvků krajiny apod. Prvotní předpoklad realizace této funkce je existující databáze obsahující informaci o prostorovém určení prvku. Právě takovou databázi nabízí služba *GeoNames*⁴⁴. Tato služba obsahuje geografická data jako názvy míst včetně jazykových variant, informace o nadmořské výšce, populaci, rozloze atp. Souřadnice těmto prvkům příslušejícím jsou uváděny v souřadnicovém systému WGS84. Velkou výhodou je bezplatnost služby, která je stejně jako *AutoSuggest* distribuována pod *Creative Commons*⁴⁵ licencí.

4.7.3 Tlačítka *smazat* a *vyhledat*

Stisk tlačítka *vyhledat* má za následek řadu reakcí. Protože je velmi problematické při funkci vyhledávání porovnávat textovou hodnotu na vstupu a hodnotu v poli *markers*, je daleko lepším řešením porovnávání unikátního *id* patřící té které skále. Aby

⁴⁴ <http://www.geonames.org>

⁴⁵ <http://creativecommons.org/licenses/by/3.0/>

to bylo vůbec možné, po výběru požadované skály ze seznamu našeptávače dojde k uložení jejího *id* do skrytého textového pole (`id='testid'`). Tato hodnota je potom načtena již známou funkcí *zoomChoice()*, blíže popsanou v kapitole 4.7.1. V případě, že funkce má na vstupu prázdnou hodnotu (tj. nedošlo k předání *id* skály skrytému poli), pod tlačítky se zobrazí informace *Skála nenalezena*. Při nalezení odpovídající si hodnoty je to zpráva *Skála zobrazena*.

Tlačítko *smazat* zavolá funkci *clearForm()*, která „vyčistí“ roletovou nabídku jmenného seznamu skal a nastaví implicitní hodnotu: *Seznam skal (A–Z)*. Současně dojde i k odstranění znaků textového pole fulltextového vyhledávání.

5 ZÁVĚR

Přestože byla volba *amapy.cz* API pro řešení mapové aplikace racionální a správná, jako drobná vada na kráse celého řešení je již zmíněný pozastavený vývoj. Mapové API je na velmi vysoké úrovni a nabízí velmi kvalitní služby, ale nereagování provozovatele na reporty chyb vyznívá poněkud rozpačitě. Dá se ale říci, že takto zásadní a komplexně zpracovaný produkt v oblasti interaktivních map nemá v rámci České republiky obdoby. I přes drobné výtky mě API po mnoha stránkách příjemně překvapilo a dá se říci, že práce s ním předčila moje očekávání. Obecně se dá předpokládat vývoj nejen mapových aplikací tímto směrem. Pomocí API se nabízí možnost programátorům a tvůrcům *mashups* využívat rozsáhlých řešení za kterými stojí týmy vývojářů a těmito principy je snadno aplikovat ve svých projektech.

Přesto by ale měl být vývojář vybaven alespoň základní znalostí objektového programování v jazyce JavaScript, znalost HTML považuji za samozřejmou. Moje předchozí zkušenosti s programováním se vztahovaly pouze na jazyk C++, HTML, ale přesto probíhala implementace mapového řešení poměrně hladce i přes nutnost seznámit se s jazykem JavaScript a vstřebat specifika tohoto vývojového prostředí. API má velmi propracovanou dokumentaci a příklady jsou logické a velmi funkční.

Řešení propojení mapové komponenty s databází pomocí ASP skriptů se projevilo jako efektivní, pouze manipulace s načtenými daty by mohla být přímočařejší. To by se dalo zajistit pomocí širšího užití formátu JSON. Zejména jeho kooperací s AJAX technologií by mohlo změnit logiku načítání POI. Jejich přidání by bylo závislé na zobrazeném výřezu oblasti mapy a tím by se jejich celkový počet na mapě při větším přiblížení snížil. Na druhou stranu by se ale zvýšilo zatížení databázového serveru,

který by byl nucen při každém posunu či změně měřítka sestavit odpověď na dotaz: „které POI do daného výřezu spadají?“. Vnitřní logika API je ale vybavena metodou, kdy sama vyhodnotí, zda-li je POI viditelný a případně negativní odpovědi se na mapě nezobrazí. I díky tomuto faktu považuji oba přístupy z hlediska efektivity za relativně vyrovnané. Větších rozdílů v rychlosti načítání a zobrazení by se dosáhlo až v případě několika tisíc POI a v tomto případě by si řešení zasloužila podrobit hlubší analýze, která se neobejde bez statistik uživatelských přístupů.

Obsah elastické bubliny a jejích záložek byl volen tak, aby umožnil rychlou orientaci a pomohl získat základní informace o skále co nejpřehledněji. Zda-li se tento záměr podařil ukáže až spokojenost uživatelů. Další možnost vylepšení a rozšíření se nabízí i v možnostech vyhledávání, jak jsem již naznačil v odstavci 4.7.2. API je možností vyhledávat ve jménech měst a krajinných prvcích nejspíše vybaveno, pouze tyto funkce nejsou plně zdokumentované.

V rozvíjení mapové aplikace bych rád pokračoval i v budoucnu a příslib administrátorů webu Skalních oblastí ČR by mi to měl umožnit.

Mapová aplikace je veřejně dostupná na webové adrese: <http://skaly.horosvaz.cz/>.

6 ZDROJE INFORMACÍ

6.1 Publikace

- [1] TYLER, Mitchell. *Web Mapping Illustrated : Using Open Source GIS Toolkits*. Sebastopol : O'Reilly Media, 2005. 349 s. ISBN 0-596-00865-1.
- [2] BRAVENEC, Vojtěch. *Přípravamapového serveru UMN pro prezentaci areálů Vysoké školy Báňské*. [s.l.], 2007. 22 s. VYSOKÁ ŠKOLA BÁŇSKÁ – TECHNICKÁ UNIVERZITA OSTRAVA. Vedoucí diplomové práce Ing. David Vojtek PhD.
- [3] WEISSINGER, A. Keyton. *Active Server Pages v kostce*. 1. vyd. Praha : Computer Press, 1999. 392 s. ISBN 80-7226-199-1.
- [4] ČERBA, Otakar. *Mapy na internetu*. Plzeň, 2006. 10 s. Západočeská univerzita v Plzni. Referát. Dostupný z WWW:
<http://gis.zcu.cz/studium/pok/Materialy/Slides/Mapy_na_internetu.pdf>.
- [5] ŘÍHA, Jan. *Distribuce map pomocí webových služeb*. [s.l.], 2007. 60 s. České vysoké učení technické v Praze. Vedoucí bakalářské práce Ing. Petr Soukup, Ph.D.
- [6] FLANAGAN, David. *JavaScript : The Definitive Guide*. 4th compl. edition. [s.l.] : O'Reilly Media, 2002. 916 s. ISBN 0596000480.

6.2 Webové stránky a články

- [url 1] JANOVSKEÝ, Dušan. *URL* [online]. [2008] [cit. 2008-04-25]. Dostupný z WWW: <<http://www.jakpsatweb.cz/html/url.html>>.
- [url 2] *Sign Up for the Google Maps API* [online]. c2008 [cit. 2008-04-25]. Dostupný z WWW: <<http://code.google.com/apis/maps/signup.html>>.
- [url 3] *Google Maps API* [online]. c2008 [cit. 2008-04-25]. Dostupný z WWW: <<http://code.google.com/apis/maps/>>.
- [url 4] CHAMPEON, Steve. JavaScript: How Did We Get Here?. *O'Reilly Network* [online]. 2001 [cit. 2008-04-30]. Dostupný z WWW: <http://www.oreillynet.com/pub/a/javascript/2001/04/06/js_history.html>.
- [url 4] ROZSYPAL, Petr. Které mapy jsou pro váš web nejlepší?. *Lupa.cz* [online]. 2007 [cit. 2008-04-11]. Dostupný z WWW: <<http://www.lupa.cz/clanky/ktere-mapy-jsou-pro-vas-web-nejlepsi/>>.
- [url 5] *Jak psát web : o tvorbě internetových stránek* [online]. [cit. 2008-04-04]. Dostupný z WWW: <<http://www.jakpsatweb.cz>>.
- [url 6] *Interval.cz : webdesign a e-komerce denně* [online]. [cit. 2008-04-21]. Dostupný z WWW: <<http://interval.cz/>>.

6.3 Originální dokumentace v elektronické podobě

- [url 7] *Google Maps API Reference* [online]. c2008 [cit. 2008-03-27]. Dostupný z WWW: <<http://code.google.com/apis/maps/documentation/reference.html>>.
- [url 8] *API Mapy.cz* [online]. c2006 [cit. 2008-04-02]. Dostupný z WWW: <<http://api.mapy.cz/>>.
- [url 9] *Atlas AMapy API : cesta k vlastním mapovým aplikacím* [online]. [2007] [cit. 2008-03-25]. Dostupný z WWW: <<http://amapy.atlas.cz/api/>>.

7 PŘÍLOHY

7.1 Obsah přiloženého CD

dp_kohout.pdf – Diplomová práce ve formátu PDF.

\horosvaz – Tato složka obsahuje veškeré soubory se zdrojovými kódy, které jsou nutné pro běh aplikace na webu Skalních oblastí ČHS.

Tento adresář dále obsahuje následující soubory a složky:

- **mapa.asp** – Hlavní soubor mapové aplikace s vygenerovaným klíčem pro doménu Skalních oblastí ČHS.
- **find.asp** – vyhledávací skript, který je volán pomocí metody AJAX
- **combo.asp** – Soubor generující detailní přehled horolezeckých výstupů vedoucích na konkrétní skálu.
- **\css** – Složka obsahující definice kaskádových stylů.
- **\js** – Složka obsahuje přídatné soubory JavaScript, na které odkazuje hlavní mapový soubor *mapa.asp*
- **\pic** – Složka s obrázky.