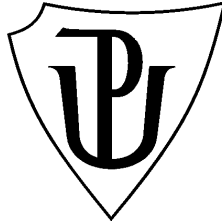


Univerzita Palackého v Olomouci
Přírodovědecká fakulta
katedra geoinformatiky



Miroslav PIZUR

**INVENTARIZACE STARÝCH
OVOCNÝCH SADŮ NA ÚZEMÍ
CHKO BÍLÉ KARPATY**

Magisterská práce

Vedoucí práce: RNDr. Vilém Pechanec, Ph.D.

Olomouc 2009

Prohlašuji, že jsem zadanou diplomovou práci řešil sám a že jsem uvedl veškerou použitou literaturu. Všechna poskytnutá vstupní a výsledná digitální data nebudu bez souhlasu školy poskytovat.

Olomouc, 27. 4. 2009

Rád bych poděkoval všem, kteří mě jakkoli podpořili při psaní této práce. Zvláštní dík patří především mému vedoucímu, RNDr. Vilému Pechancovi, Ph.D., za pomoc s výběrem tématu, kontinuální podporu a inspirativní návrhy.

Vysoká škola: Univerzita Palackého v Olomouci

Fakulta: Přírodovědecká

Katedra: Geoinformatiky

Školní rok: 2006/2007

ZADÁNÍ DIPLOMOVÉ PRÁCE

pro Miroslava PIZURA

obor Aplikovaná geoinformatika

Název tématu:

**WEBOVÝ PORTÁL PRO INVENTARIZACI STARÝCH OVOCNÝCH SADŮ
NA ÚZEMÍ BÍLÝCH KARPAT
WEBSITE FOR OLD ORCHARD INVENTORY IN BILE KARPATY AREA**

Zásady pro vypracování:

Cílem diplomové práce je vytvoření webového portálu, který umožní inventarizaci starých ovocných sadů. Portál musí nabízet nástroje pro založení nového záznamu, včetně jeho lokalizace, správu a prezentaci obsažených dat. Portál musí umět řešit jak prostorovou, tak i popisnou část inventarizačního záznamu. Autor se bude snažit o řešení, které bude mít minimální nároky na vlastní hardware, software a pořízení podpůrných dat. Systém bude naplněn dodanými daty. Při realizaci bude autor respektovat pravidla pro tvorbu webových aplikací a výsledný produkt bude zcela založen na nekomerčních řešeních.

Postup prací

- sběr a zpracování podpůrných dat pro inventarizaci (číselníky, formuláře, ...),
- tvorba konceptu webového portálu,
- zpracování projektové dokumentace,
- testování pilotní verze portálu a naplnění vstupními daty,
- tvorba finální verze mapového portálu a jeho spuštění .

O diplomové práci student vytvoří internetovou stránku, která bude v den odevzdání práce umístěna na server UP. Na závěr práce připojí jednostránkové resumé v anglickém jazyce. Výstupy budou odevzdány v digitální podobě na CD – ROM. Student odevzdá údaje o všech datových sadách, které vytvořil nebo získal v rámci práce, pro potřeby zaevidování do Metainformačního systému katedry geoinformatiky ve formě vyplněného dotazníku. Práce bude zpracována podle zásad dle Voženílek (2002).

Rozsah grafických prací:

série schémat, map a grafů dle potřeby práce

Rozsah průvodní zprávy:

min. 50 stran textu

Seznam odborné literatury:

- Burian, J. (2005): *Internetové řešení územního plánu města Náměště nad Oslavou. [Bakalářská práce] Katedra geoinformatiky, PřF, Univerzita Palackého v Olomouci.*
- Grusová., L. (2002): *XML pro úplné začátečníky. Praha, Computer Press, 2002 196 s. ISBN 8072266977*
- Mitchel, T. (2005): *Web mapping illustrated. Using Open Source GIS Toolkits. Sebastopol, O Reilly Media., 349 s.*
- Peng, Z-R., Tsou, M-H. (2003): *Internet GIS : distributed geographic information services for the Internet and wireless networks. Hoboken, Wiley and Sons, 2003 679 s. ISBN 0471359238*
- Spainhour, S., Eckstein, R. (1999): *Webmaster v kostce. Praha, Computer Press., 503 s.*
- Tang, W., Selwood, J. (2003): *Connecting our world: GIS Web services. Redlands, ESRI, 2003, 164 s. ISBN 1589480759*
- Voženílek, V. (2002): *Diplomové práce z geoinformatiky. Vydavatelství Univerzity Palackého, Olomouc, UP, 31 s.*
- Welling, L., Thomson, L. (2002): *PHP a MySQL - rozvoj webových aplikací. Praha, SoftPress, 718 s. ISBN 8086497208*

URI

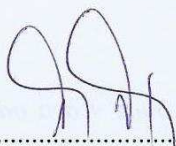
<http://mapserver.gis.umn.edu/>

<http://maps.google.com/>

Vedoucí diplomové práce: Mgr. Vilém Pechanec, Ph.D.

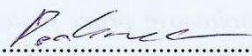
Datum zadání diplomové práce: 4. prosince 2006

Termín odevzdání diplomové práce: 30. dubna 2008



Vedoucí katedry

UNIVERZITA PALACKÉHO
771 46 OLOMOUC, tř. Svobody 26
Přírodovědecká fakulta
Katedra geoinformatiky
tel. 585 634 513, 585 634 516 ②



Vedoucí diplomové práce

V Olomouci dne 4. prosince 2006

OBSAH

ÚVOD	- 8 -
1. CÍL PRÁCE.....	- 9 -
2. SEZNAM ZKRATEK.....	- 10 -
3. WEBOVÉ SLUŽBY	- 11 -
3.1. Využití webových služeb	- 11 -
3.2. Webové služby v oblasti GIS	- 12 -
3.2.1. Specifikace OGC webových služeb	- 12 -
3.2.1.1. Web map services.....	- 12 -
3.2.1.2. Web feature services	- 13 -
4. TECHNOLOGIE.....	- 14 -
4.1. (X)HTML	- 14 -
4.2. Kaskádové styly CSS	- 15 -
4.3. JavaScript	- 16 -
4.4. Klientský JavaScript.....	- 17 -
4.5. DOM.....	- 17 -
4.5.1. Procházení DOM stromem.....	- 18 -
4.5.2. Změny DOM stromu	- 18 -
4.5.3. DOM události	- 18 -
4.6. AJAX.....	- 20 -
4.6.1. Termín AJAX.....	- 20 -
4.6.2. Model webové aplikace.....	- 21 -
4.6.3. Využití AJAXu.....	- 22 -
4.6.4. Stinné stránky AJAXu.....	- 23 -
5. API MAPOVÁ TECHNOLOGIE	- 24 -
5.1. Filozofie API.....	- 24 -
5.2. Mashup – možnost využívat API a vytvářet nové služby	- 24 -
6. API TUZEMSKÝCH A ZAHRANIČNÍCH POSKYTOVATELŮ	- 28 -
6.1. České	- 29 -
6.1.1. Seznam	- 29 -
6.1.2. Atlas	- 29 -
6.1.3. Vzájemné porovnání	- 30 -
6.1.4. Rychlost načítání značek.....	- 30 -
6.1.5. Celkové zhodnocení (Seznam x Atlas)	- 31 -
6.2. Zahraniční.....	- 32 -
6.2.1. Srovnání mapových dat pro Českou Republiku.....	- 33 -
6.2.2. Celkové zhodnocení	- 34 -
7. CHARAKTERISTIKA ZÁJMOVÉHO ÚZEMÍ	- 35 -
8. GOOGLE MAPS API	- 38 -
8.1. API klíč	- 38 -
8.2. Licenční podmínky.....	- 38 -
8.3. Dokumentace Google API	- 38 -
8.4. Základní mapové objekty	- 39 -
8.5. Úskalí aplikace	- 39 -
8.6. Nejzajímavější aplikace vytvořené nad mapovými API (nejen Google)	- 40 -

9.	METODY A POSTUP ZPRACOVÁNÍ	- 44 -
9.1.	Výběr prostředí.....	- 44 -
9.2.	Inspirace	- 44 -
9.3.	Data a jejich konverze	- 45 -
9.4.	Aplikace	- 45 -
9.5.	Sepsání dokumentace	- 46 -
10.	POSTUP PRÁCE	- 47 -
10.1.	Výběr API mapového prostředí.....	- 47 -
10.2.	Data	- 48 -
10.3.	Vytvoření aplikace	- 50 -
10.3.1.	Použité nástroje:	- 51 -
10.3.2.	Filosofie aplikace	- 51 -
10.4.	Testování aplikace.....	- 53 -
10.5.	Výsledná aplikace - co obsahuje a jak pracuje.....	- 55 -
10.5.1.	Generalizace	- 55 -
10.5.2.	Legenda	- 58 -
10.5.3.	Výpis záznamů	- 58 -
10.5.4.	Google Earth	- 59 -
10.5.5.	Tisk.....	- 59 -
10.5.6.	Info o aplikaci.....	- 60 -
10.5.7.	Přidání externí vrstvy	- 60 -
10.5.8.	Přes celé okno.....	- 61 -
10.6.	Editační prostředí	- 63 -
10.7.	Ovládání aplikace	- 64 -
10.8.	Diskuse o nekompatibilitě prohlížečů - co musí být nastaveno.....	- 65 -
11.	ZÁVĚR.....	- 66 -
12.	LITERATURA.....	- 68 -
13.	SUMMARY	- 72 -
	SEZNAM PŘÍLOH	- 73 -

ÚVOD

Díky technologickému pokroku v GIS opensourcových technologiích, obecně se zvyšující rychlosti připojení k internetu a v neposlední řadě rozšiřujícím se povědomím a zájmem o počítačovou kartografii mezi širokou laickou veřejností, vznikl dnes poměrně nový technologický prvek. Možnost zadarmo si vytvořit vlastní GIS aplikaci.

Myšlenka vytvoření vlastního GISu za nulové finanční náklady je již staršího data. Avšak až během posledních let přišli některé na internetovém trhu významné společnosti s nápadem, jak částečně „eliminovat“ v této oblasti asi nejproblémovější místo. Data. Můžeme mít software propracovaný do sebemenšího detailu, nicméně bez kvalitních dat stejně žádný relevantní výstup nevytvoříme. Obecně platí, že data tvoří 80 % finančních nákladů celkového GISu... Asi by nebylo správné tvrdit, že tímto krokem se u takto postavených aplikací problém dat zcela vyřešil. Velká většina geoinformatických projektů se stejně bez dalších přídavných dat neobejde. Nicméně minimálně pro propagační, školní a mnoho jiných, většinou nekomerčního charakteru, tvůrčích nápadů, toto řešení bohatě postačuje.

Zmiňované typy projektů využívají na internetu mapy spíše jako podkladovou vrstvu, nikoli jako vrstvu pro následné analýzy. A právě ta jsou dnes společnostmi zdarma k dostání. Jedná se o rastrová data pokrývající celý svět. Mapy jsou jak standardní topografická, tak satelitní o velkém rozlišení. Díky programovatelnému rozhraní API je možno k nim vzdáleně přistupovat a na nich následně vytvářet přídavné moduly. Jedinou logicky zbývající podmínkou pro vytvoření aplikace je připojení k internetu.

Výsledná aplikace je záležitostí čistě dobrého nápadu a jeho schopnosti ho programátorsky zrealizovat.

1. CÍL PRÁCE

Stěžejním cílem diplomové práce je vytvoření webového portálu, který umožní inventarizaci starých ovocných sadů. Aplikace bude zaznamenávat do mapy přesné polohy jak ovocných stromů, tak sadů v lokalitě CHKO Bílé Karpaty. Tj. musí umožňovat vkládání záznamu, jeho smazání, editaci i samotné zobrazení v mapě. Ke geografické informaci musí být možnost připojit i údaj popisný, taktéž libovolně editovatelný.

Práce se dá rozdělit zhruba do čtyř částí. Prvotní spočívá ve výběru vhodného prostředí pro zpracování tohoto úkolu. Druhá ve vytvoření samotné aplikace a třetí ve zpracování její dokumentace. Posledním krokem je nasazení a následné seznámení této aplikace s pracovníky CHKO Bílých Karpat.

O diplomové práci bude vytvořena internetová stránka, která bude umístěna na serveru UP.

2. SEZNAM ZKRATEK

AJAX Asynchronous JavaScript and XML
API Application Program Interface
CSS Cascading Style Sheets
DOM Document Object Model
ECMA European Computer Manufacturers Association
ERD Entity-relationship diagrams
GIS Geographic information system
HTTP Hyper Text Transfer Protocol
HTML HyperText Markup Language
IE Internet Explorer
MF Mozilla Firefox
OGC Open Geospatial Consortium
RIA Rich Internet applications
SMTP Simple Mail Transfer Protocol
SOAP Simple Object Access Protocol
SQL Structured Query Language
TCP Transmission Control Protocol
UDDI Universal Description Discovery and Integration
URI Uniform Resource Identifier
URL Uniform Resource Locator
W3C World Wide Web Consortium
WCS Web Coverage Service
WFS Web Feature Service
WMS Web Mapping Service
WSDL Web Services Description Language
WWW World Wide Web
XHTML Extensible Hypertext Markup Language
XML Extensible Markup Language

3. WEBOVÉ SLUŽBY

Zpracováno dle Richarda Ondry [6].

Webové služby představují posun od velkých monolitních struktur aplikací k modelu založenému na komponentech. Aplikace jsou v rámci tohoto modelu sestavené z malých stavebních prvků - jednotlivých funkcí. Pokud jsou tyto funkce umístěné na různých internetových serverech, označují se jako webové služby. Takto sestavené aplikace je možné snadno vytvořit, dynamicky modifikovat a měnit.

Webová služba je souhrnný název pro skupinu technologií a metod, které spojují informační systémy prostřednictvím internetu a umožňují jim spolu efektivně komunikovat. Webové služby s použitím XML pro výměnu dat dovolují spojit různé aplikace bez ohledu na počítačovou platformu, použité programovací jazyky a síťové protokoly [W27].

3.1. Využití webových služeb

Webové služby mohou být využity nejen klientskými či webovými aplikacemi, ale také jinými webovými službami. Na straně klientů mohou být jak klasické osobní počítače tak i zařízení typu PDA nebo mobilní telefony. Pro programátora je způsob využívání webových služeb velmi blízký způsobu využívání tříd.

Webové služby je možné spolu s dalšími komponenty vyhledávat v registrech a vytvořit z nich aplikace. Při spojování webových služeb je však důležité propojovat služby inteligentně, to znamená tak, aby výsledná síť webových služeb fungovala v rámci procesních a obchodních pravidel. Dále je nutné zajistit runtime provozní prostředí, které je vysoce spolehlivé, bezpečné a je ho možné spravovat lokálně i vzdáleně. Takové prostředí je postavené na aplikačním serveru. Ten, kdo tuto síť tvoří, je IT manažer nebo obchodní analytik zaměřený spíše na obchodní procesy než na programování. Využívá k tomu nástroje ke znázornění obchodního procesu a formulaci nezbytné procesní logiky bez toho, že by musel psát nějaký kód. Většinu z těchto pravidel přitom určují obchodní analytici a ne programátoři [W27].

Ve chvíli, kdy jsou obchodní procesy vytvořené, jsou jednotlivým koncovým uživatelům poskytnuty různými způsoby (například přes prohlížeč, mobilní telefon nebo mainframe).

3.2. Webové služby v oblasti GIS

V poslední době stále více dochází k přechodu řady aplikací z prostředí „desktopových“ produktů na Internet do prostředí distribuované správy. Tento trend se týká i GIS, kde dochází ke vzniku internetových mapových aplikací, které využívají webové mapové služby. Nejčastěji se jedná o služby definované otevřenými standardy Open GIS konsorcia. Z těchto standardů jsou nejčastěji používány Web Map Services, Web Feature Services.

Hlavním přínosem webových mapových služeb definovaných dle Open GIS konsorcia [W28] je umožnění sdílení dat GIS v distribuovaném prostředí Internetu. Uživatelé tím mohou sdílet mapy a aplikace bez nutnosti mít příslušná data na svém počítači nebo serveru. Typickým příkladem je zobrazení komplexní tematické mapy obsahující data z různých serverů on-line v internetovém prohlížeči (tenkém klientovi) nebo v nějakém desktopovém GIS programu (tlustém klientovi).

3.2.1. Specifikace OGC webových služeb

Podrobný přehled specifikací OGC pro webové služby lze nalézt na stránkách konsorcia (www.opengis.org). Zde se zmíním z celé řady služeb pouze o dvou výše jmenovaných, které se problematiky bezprostředně týkají:

3.2.1.1. Web map services

Slouží k poskytování map, které jsou generovány dynamicky z geografických dat. Mapou se rozumí vizualizace dat, nikoliv data samotná. WMS tedy na základě požadavku připraví rastrový obrázek, který poskytne klientovi. Webová mapová služba je identifikována URL. Požadavky na službu předává klient pomocí parametrů v adrese. WMS na základě požadované operace vrátí buď obrázek s mapou nebo jinou požadovanou informaci (metadata, chybové hlášení) nejčastěji v jazyce XML [W28].

Standard definuje tři operace:

- **getCapabilities:** vrací XML dokument popisující celou službu (seznam vrstev, informace o formátu mapy, podporovaných kartografických projekcích, způsob ošetření výjimek...). Aplikace z tohoto dokumentu čtou informace pro další spolupráci se serverem.
- **getMap:** vrací mapu ve formě obrázku (GIF, PNG, ...).
- **getFeatureInfo:** vrací atributy prvku mapy na souřadnicích zadaných uživatelem (text, GML, HTML apod.)

Klientem WMS může být libovolný webový prohlížeč a parametry může uživatel serveru předávat přímo v URL. Mnohem častěji je klientem specializovaná aplikace (i webová), která poskytuje uživateli grafické

rozhraní k zadávání parametrů. Takováto aplikace pak může zároveň kombinovat mapy z různých zdrojových serverů WMS. Pokud klientem bude aplikace, která je zároveň WMS serverem může poskytovat mapy vytvořené kombinací map z různých jiných serverů. To umožňuje vytvořit síť distribuovaných mapových serverů.

3.2.1.2. Web feature services

Služba je určena k přenosu vektorových dat (ve formátu GML). Z důvodu množství přenášených dat se hodí spíše pro práci s vlastními daty nebo např. při analýzách. Práce s GML ovšem vyžaduje tlustého klienta [W28].

WFS definuje tyto operace:

- getCapabilities – slouží pro získání metadat o službách poskytovaných serverem
- WMS,
- getFeature – vrací geografická data objektů na základě specifikovaných geografických či atributových vlastností,
- getFeatureType – poskytuje detailní informace o objektech, insertFeature
- (volitelné) – pro vložení objektu do systému,
- updateFeature (volitelné) – pro aktualizaci objektu,
- deleteFeature (volitelné) – pro odstranění objektu ze systému.

4. TECHNOLOGIE

Zpracováno dle Richarda Ondry [6].

Tato kapitola popisuje základní stavební kameny uživatelského rozhraní mapových aplikací vznikajících na základě využívání API mapových služeb. Základní uživatelské rozhraní se skládá z XHTML, CSS pro definici vzhledu a obsahu, JavaScriptu pro detekci událostí, změn na stránce a další programové logiky.

4.1. (X)HTML

HTML je klíčovým značkovacím jazykem pro tvorbu webových stránek. Popisuje strukturu webové stránky, obsahuje text, kterému dává různé typy včetně hypertextových odkazů, a dovoluje tvořit formuláře, obrázky a další objekty. Na základě HTML vznikl jazyk XHTML, který se dá chápat jako mírná úprava HTML při zachování jeho funkčnosti do takové formy, aby splňoval pravidla daná pro formát XML. Validní XHTML tak může být zpracováváno automatickými nástroji pro XML. Změnou oproti HTML je dle [10], že dokument musí být „dobře formátovaný“ podle požadavků na XML:

- všechny značky musí být ukončeny,
- ve jménech atributů se rozlišuje velikost písmen (v XHTML se píší malými písmeny),
- všechny hodnoty atributů musí být v uvozovkách (ať jednoduchých nebo dvojitéch),
- a další méně významná pravidla specifikovaná w3c.

Protože XHTML dokument musí tvořit strom, je nutné správně vnořovat elementy. To znamená, že se při zápisu nemohou překrývat jako například: `Odkaz1 - Popis`

V takovém případě by strom zpracováním XHTML dokumentu nemohl vzniknout. Stránka by sice pravděpodobně fungovala, ale každý prohlížeč přeformátovává špatně formátované dokumenty po svém. Validitu dokumentu je možné ověřit validátory zabudovanými v některých vývojových prostředích nebo pomocí webových validátorů (validator.w3.org).

XHTML bývá po žádosti uživatele generováno na serveru skriptovacím jazykem nebo poskytováno už jako předgenerovaný statický soubor. Ajax (viz. 5.6. AJAX) umožňuje i případ, kdy XHTML stránka, která dorazí do prohlížeče, obsahuje jen hlavičku a prázdné tělo. Obsah stránky je vygenerován JavaScriptem z dat dodaných při nahrání a doplňovaný o data

dorazivší asynchronně. Rady opakující se od zkušených vývojářů říkají, že moderní webové stránky by měly definovat strukturu stránky v XHTML, zatímco formátování, barvy, pozice by měly být definovány v pomoci CSS.

Dalším trendem v posledních letech byl přechod od stránek, jejichž vzhled byl složen z mnohonásobně vnořených tabulek k stránkám, kde jsou jednotlivé oblasti tvořeny elementy div, jejichž pozice, ať už relativní nebo absolutní, je definována opět v CSS a může být dynamicky měněna JavaScriptem. Dodržování těchto pravidel je výhodné i v mapových aplikacích, protože zpřehledňuje kód, zvyšuje znouvopoužitelnost kódu a snižuje složitost manipulace s objektovým modelem dokumentu [W29][10].

4.2. Kaskádové styly CSS

CSS je jazyk popisující prezentaci dokumentu zapsaného pomocí značkovacího jazyku, například (X)HTML a XML. Lze s jeho pomocí měnit velikost a typ písma, barvy, orámování a dokonce i umístění elementů.

Jazyk byl navržen organizací W3C. Zatím byly vydány dvě verze specifikace CSS 1, CSS 2 a revize CSS 2.1. V současné době se pracuje na verzi CSS 3. Některé předpokládané vlastnosti verze 3 jsou již ze strany prohlížečů podporovány, avšak v současné době se jedná téměř výhradně o prohlížeč Firefox. Hlavním smyslem je umožnit návrhářům oddělit vzhled dokumentu od jeho struktury a obsahu. Původně to měl umožnit už jazyk HTML, ale v důsledku nedostatečných standardů a konkurenčního boje výrobců prohlížečů se vyvinul jinak. Starší verze HTML obsahují celou řadu elementů, které nepopisují jenom obsah a strukturu dokumentu, ale i způsob jeho zobrazení. Z hlediska zpracování dokumentu a vyhledávání informací není takový vývoj žádoucí [11][W30].

Používání kaskádových stylů ve srovnání se samotným HTML v praxi přináší dle [11] následující výhody:

- větší formátovací možnosti,
- snadná tvorba a údržba konzistentního stylu,
- oddělení struktury a stylu,
- dynamická práce se styly,
- formátování XML dokumentu,
- větší kompatibilita alternativních webových prohlížečů,
- kratší doba načítání stránky.

Obrovskou výhodou CSS oproti starému formátování v HTML je, že kód a obsah webu je uložen v jednom souboru a veškerý design a formátování se

načítá ze souboru (*.css). Tento soubor je často společný pro celý web. Stylový předpis se skládá z posloupnosti pravidel. Každé pravidlo určuje vzhled některého elementu dokumentu nebo skupiny elementu. Pravidlo začíná tzv. selektorem, který specifikuje skupinu elementu. Selektor je následován seznamem deklarácí, které určují vzhled vybrané skupiny elementů. Celý seznam je uzavřen ve složených závorkách a jednotlivé deklarace jsou odděleny středníkem [W31].

4.3. JavaScript

JavaScript je interpretovaný programovací jazyk se základními objektově orientovanými schopnostmi. Univerzální jádro jazyka bylo vloženo do Mozilly (Netscape Navigatoru), Internet Exploreru a dalších webových prohlížečů. Pro webové programování bylo také rozšířeno přidáním objektu reprezentujícího okno webového prohlížeče a jeho obsah. Tato klientská verze JavaScriptu umožňuje vložit do webových stránek proveditelný obsah – to znamená, že stránka na webu již nemusí být jen statickým dokumentem HTML, ale může obsahovat dynamické programy, které komunikují s uživatelem, řídí prohlížeč a dynamicky vytvářejí obsah HTML [12].

Jádro jazyka JavaScript svými programovými konstrukcemi, jako je příkaz if, cyklus while a operátor &&, syntakticky připomíná programovací jazyky C, C++ a Javu. Avšak syntaxí podobnost končí. JavaScript je programovací jazyk bez typové kontroly, což znamená, že proměnné nemusí mít specifikovaný typ. Objekty v JavaScriptu se podobají spíše asociativním polím jazyka Perl, než strukturám v C nebo objektům v C++ a v Javě.

Mechanismus dědičnosti objektové orientace JavaScriptu odpovídá prvkům málo rozšířených jazyků Self a NewtonScript a výrazně se liší od dědičnosti v C++ a Javě. Podobně jako Perl je JavaScript interpretovaný jazyk a mnohé myšlenky přebírá z jazyka Perl. Příkladem jsou jeho regulární výrazy a nástroje pro práci s poli. Ačkoli nemají JavaScript a Java mnoho společného, často se na webu využívá „spolupráce“ těchto jazyků. Tyto dva jazyky mají odlišné sady schopností. JavaScript může řídit chování a obsah prohlížeče, ale nemůže kreslit grafiku nebo pracovat se sítí. Naopak Java má vlastnosti opačné. Java nemá žádnou kontrolu nad prohlížečem jako celkem, ale může vytvářet grafiku a pracovat se sítí.

Klientský JavaScript může pracovat s applety Javy vloženými do webové stránky a řídit je. Touto kombinací lze získat mocný nástroj pro webové programování [13].

4.4. Klientský JavaScript

Když je překladač JavaScriptu vložen do webového prohlížeče, výsledkem je klientský JavaScript. Toto je zdaleka nejběžnější forma JavaScriptu. Většina lidí, pokud se zmiňuje o JavaScriptu, má obvykle na mysli klientský JavaScript. Ten kombinuje možnost zpracování skriptu interpretem JavaScriptu s objektovým modelem dokumentu (DOM – viz kap. 4.5) definovaným webovým prohlížečem. Tyto dvě odlišné technologie se synergicky doplňují. JavaScript umožňuje distribuci spustitelného obsahu přes web a je srdcem dokumentu dynamického HTML. Podobně jako specifikace ECMA definuje standardy jazyka JavaScript, tak organizace W3C publikovala specifikaci DOM určující prvky, které musejí prohlížeče ve svém DOM podporovat [13].

4.5. DOM

DOM objektový model dokumentu je vlastně aplikační programové rozhraní. Pomocí tohoto rozhraní můžeme pracovat se samotnou webovou stránkou. DOM umožňuje přistupovat k jednotlivým objektům XHTML dokumentu a pracovat s nimi. Těmito objekty jsou elementy, atributy, text, komentáře... A dovede to prostřednictvím JavaScriptu. Pro tvorbu mapových aplikací nad API vybraných mapových služeb je tedy nepostradatelný.

Díky stromové struktuře dokumentu, dovedeme rozlišit nadřazené nebo podřazené elementy - `document.form.name`, objekt `form` je podřazený objektu `document`. Každý objekt může mít vlastnosti a metody. Vlastnost sama o sobě nedělá v dokumentu žádné změny, pouze vrací informace o objektu. Třeba `form.name` vrací jméno formuláře. Oproti tomu metody umožňují data nějakým způsobem měnit, nebo provádět nejrůznější akce. Kupříkladu `form.submit()`, odesílá formulář. Jestli jde o vlastnost nebo metody lze rozpoznat podle závorek (metoda má závorky, vlastnost nemá). Ač už jde o metodu nebo o vlastnost je spojena s objektem tečkovým zápisem.

V devadesátých letech byl DOM v prohlížečích implementován různě, bez společného standardu. Tento stav ztěžoval vývoj webových stránek. Později byl DOM standardizován W3C, i když společná podpora v prohlížečích se zlepšila jen pozvolna. Současná verze z roku 2004 nese označení DOM Level 3. Kompletní podpora v prohlížečích je ale jen pro DOM Level 1 a větší část specifikace DOM Level 2 [W41].

4.5.1. Procházení DOM stromem

JavaScript přistupuje ke kořeni dokumentu tvořícího právě načtenou stránku přes globální proměnnou `document`, která slouží jako základ pro všechny manipulace DOM. Každý uzel má jednoho rodiče a libovolné množství potomků. Libovolné množství atributů každého uzlu je uloženo v asociativním poli [13].

JavaScript může načíst element několika způsoby jak uvádí:

- Metodou `document.getElementById(<id uzlu v XHTML>)`. Takto načte jeden konkrétní element určený jedinečným identifikátorem `id`. Atribut `id` je elementu přiřazen buď přímo v XHTML dokument, nebo už modifikací DOM JavaScriptem.
- Metodou `document.getElementsByTagName(<jméno tagu>)`. Výstupem metody je pole se všemi elementy stejného jména, například všechny odkazy nebo obrázky. Není tedy nutné předtím přiřadit elementům identifikátory.
- Procházením stromem. Pokud je známa struktura DOM stromu, je možné získat například přístup k rodiči (`element.parentNode`), potomkům (`element.childNodes`), sourozencům (`element.nextSibling`). Předpokladem je existence alespoň jednoho už načteného elementu, od kterého se odvíjí další procházení stromu. V praxi je to často element `body` (tělo dokumentu), který se získá vyhledáním podle jména `body=document.getElementsByTagName('body')`.

4.5.2. Změny DOM stromu

Prostřednictvím JavaScriptu můžeme elementy:

- Vytvářet: `document.createElement(<jméno elementu/tagu>')`. Po vytvoření je element umístěn jen v dočasné proměnné a musí se ještě umístit na správné místo v DOM stromu: `<budoucí rodič>.appendChild(<před chvílí vytvořený element>)`.
- Mazat: `<element ke smazání>.parentNode.removeChild(<element ke smazání>)`. Odstranění je tedy aplikováno na rodiče jako smazání jeho potomka. Smazání elementu smaže i všechny jeho potomky a je vhodné pro úplné odstranění (někdy je vhodnější skrytí) elementu ze stránky, a případné nahrazení jinými, které obsahují požadovaná data.
- Přesouvat: `<nový rodič>.appendChild(<element pro přesunutí>)` Přesunout část stránky je takto rychlejší, než smazání a opětovné vytvoření podstromu.
- Měnit jejich hodnoty: `element.nodeValue`.

4.5.3. DOM události

V případě, že se elementu stane nějaká akce z předem definované množiny pak je spuštěna DOM událost. Může to být kliknutí myši nebo třeba změna formuláře. Události způsobuje zpravidla uživatel, ale také mohou být generované prohlížečem (například událost načtení nějakého objektu).

Události, které jsou standardizované ve specifikaci DOM Level 2 a zároveň jsou podporovány všemi prohlížeči, uvádí následující tabulka.

Tab. 1 Přehled DOM událostí [8]

Kategorie	Název	Elementy	Událost nastane
Myš	Click	všechny viditelné	uživatel klikne na jakékoliv tlačítko, když je myš nad elementem. Klik je definován jako zmáčknutí tlačítka a uvolnění.
	dblclick	všechny viditelné	při dvojkliku na element
	Mousedown	všechny viditelné	při zmáčknutí tlačítka myši
	Mouseup	všechny viditelné	po uvolnění tlačítka myši
	mouseover	všechny viditelné	při umístění kurzoru nad element
	mouseout	všechny viditelné	když kurzor opustí element
	mousemove	všechny viditelné	při pohybování myši nad elementem
Klavesnice	keypress	elementy s focusem	při stisknutí a následovném uvolnění tlačítka Klavesnice
	Keydown	elementy s focusem	při Stisku tlačítka na klavesnici
	KeyUp	elementy s focusem	při uvolnění tlačítka
HTML rámeček nebo objekt	Load	window, frameset	po dokončení načítání stránky nebo elementu.
	unload	window, frameset	po odebrání celého obsahu stránky z okna (například při zavření okna nebo přechodu na jinou stránku)
	abort	object	při přerušení před dokončením načtení.
	error	object, body, frameset	když nemůže být obrázek, objekt nebo rámeček načten.
	resize	windows	při změně velikosti okna.
	scroll	Windows	při skrolování
HTML formulář	select	textové pole	při výběru textu
	Chase	vstupní pole	když je vstupní pole deaktivováno a bylo od doby aktivace změněno.
	submit	formulář	při odeslání formuláře.
	reset	formulář	když je formulář resetován.
	focus	label, input, select, textarea, button	při aktivaci elementu.
Blur	label, input, select, textarea, button	při deaktivaci elementu.	

Události spouštěné myší, jsou v praxi využívány nejvíce, protože také uživatelé k ovládání webových stránek a online aplikací používají nejčastěji myš. Z atributů objektu spuštěné události je možné zjistit kontextové informace jako pozici kurzoru, zda byla stisknuta klávesa a jiné. Sledování pohybu myši, mačkání tlačítek a reakce na ně jsou nutné pro přiblížení rozhraní webu desktopovým aplikacím. Standardním HTML prvkům je tak možné přidat další funkcionalitu a je možné vytvářet úplně nové jako

virtuální okna. Tyto události jsou nezbytné pro mapové aplikace z důvodů zjišťování pozice kurzoru nad mapou, umístování značek na mapu nebo zjišťování aktivity uživatele.

Díky událostem, které jsou spouštěné klávesnicí, je možné vytvořit vlastní ovládání aplikace klávesami a klávesovými zkratkami. Dle Usability Testing [W32] je ovládání aplikace klávesnicí mnohdy rychlejší, než kombinací myši a klávesnice.

Vhodné je inicializovat JavaScriptové aplikace událostí load, kterou přiřadíme tělu dokumentu – tagu body. Jen tak je zajištěno, že elementy, kterým chceme obsluhu událostí přiřadit, jsou už načteny v prohlížeči. Jedná se o událost spuštěnou dokumentem. Dalšími událostmi tohoto typu jsou resize a scroll, které dávají možnost přizpůsobit se změně v pozici dokumentu.

Události spouštěné HTML formuláři umožňují zvýšit interaktivitu při zadávání dat a rozšířit funkcionalitu formulářů.

4.6. AJAX

V této kapitole bude podrobněji probrána technologie AJAX, kterou využívají portály Seznam i Atlas pro dynamické načítání obrázků mapy ve svých mapových aplikacích. Konceptu AJAX bylo také využito pro předávání informací, generovaných skriptem na straně serveru, do klientského JavaScriptu, ve vytvořené on-line GIS aplikaci.

4.6.1. Termín AJAX

Zkratka AJAX znamená Asynchronous JavaScript and XML. AJAX není sám o sobě implementací technologie či softwarovým produktem, ale jedná se o obecný koncept nebo lépe návrhový vzor pro RIA. AJAX je představitel cesty využívající maximální možné hodnoty dnešních technologií. AJAX je technika sestávající se dle [14] z následujících technologií:

- Document Object Model,
- XMLHttpRequest,
- HTML, CSS,
- JavaScript,
- XML

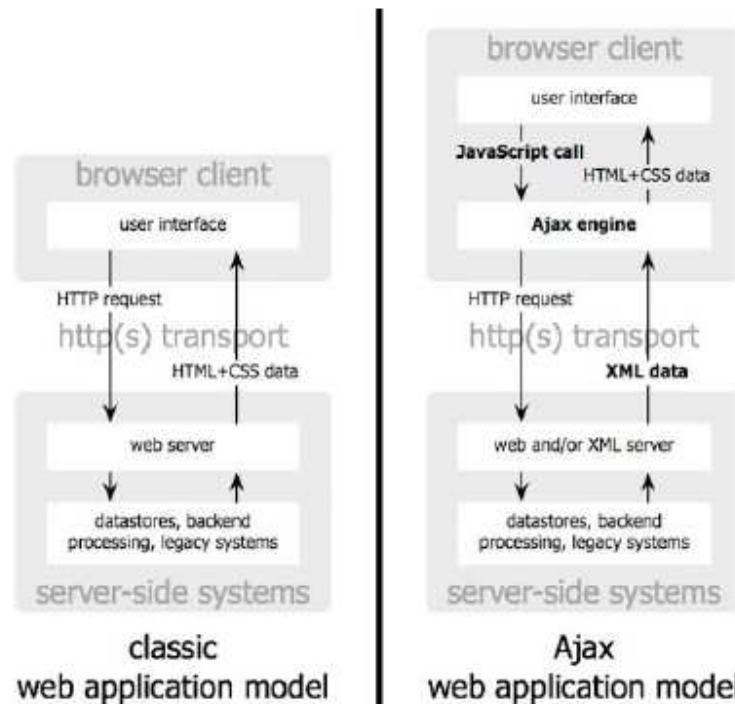
AJAX je tedy konceptem využití těchto technologií, které byly výše popsány. Cílem Ajaxu je zvýšit interaktivitu, rychlost a použitelnost webových stránek. Ve vysoce konkurenčním prostředí světového internetu se cení každý detail a každá ušetřená sekunda času uživatele. V současné době je Ajax čím dál rozšířenější, v České Republice je jeho nasazení trochu pozadu, ale například

portál Seznam před časem implementoval našeptávač, který stojí na této technice.

4.6.2. Model webové aplikace

Ve standardním modelu webové aplikace, je browser zodpovědný za iniciaci žádosti webovému serveru a její zpracování. AJAX model k tomu využívá mezivrstvy – Ajax Engine. Ajax engine je obvykle JavaScriptový objekt nebo funkce která je volána, kdykoli mají být od serveru získány nějaké informace.

Server, který dodává HTML dokumenty, CSS šablony vzhledu nebo JavaScriptový kód, je konfigurován tak, aby vracel data, která může Ajax engine zpracovat. Tato data mohou být například ve formě obyčejného textu, XML dokumentu nebo čehokoliv co umí Ajax engine interpretovat. V okamžiku obdržení odpovědi může Ajax engine provést příslušnou akci. Oproti klasickému webovému aplikačnímu modelu, tento proces zahrnuje přenos menšího množství informací, což znamená, že bude docházet k rychlejší aktualizaci. Porovnání standardního webového modelu s modelem Ajaxu je ilustrováno na Obr. 1. [W16]

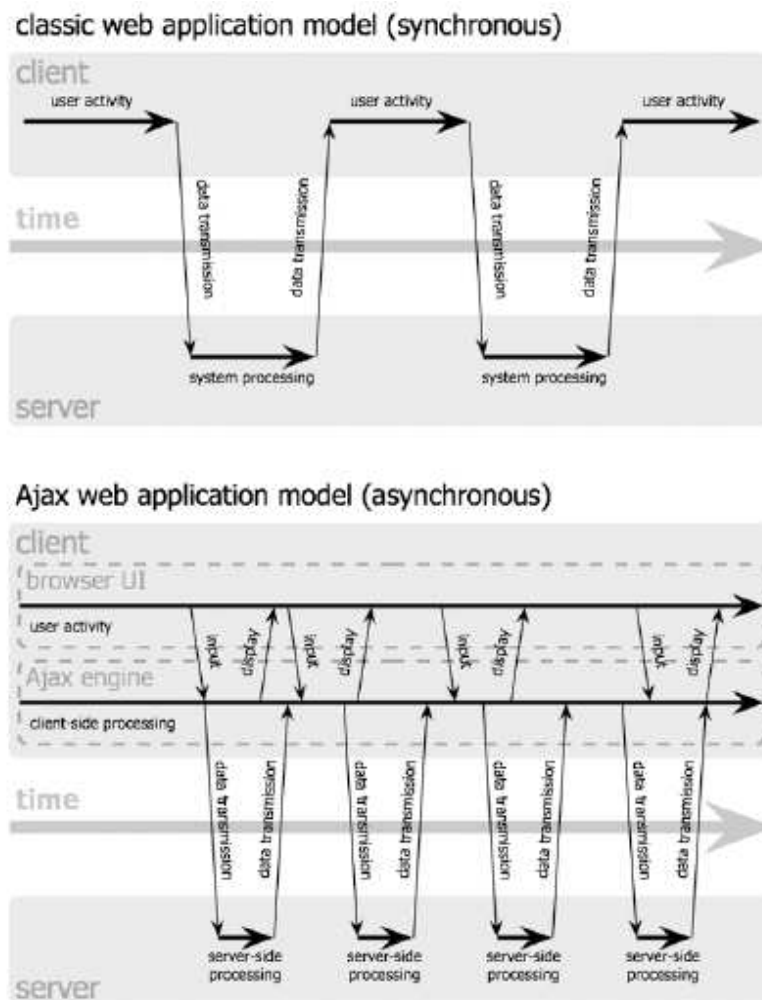


Obr. 1 Porovnání klasického modelu s modelem AJAXu [W8]

Základním stavebním kamenem je objekt XMLHttpRequest, který umožňuje asynchronní volání serveru. V klasickém webovém modelu každá změna stavu na klientovi vyžaduje obnovení celého uživatelského rozhraní. Vše

probíhá v pevně dané posloupnosti kroku. Nejdříve je vygenerována žádost o změnu stavu, pak dochází k odeslání požadavku na server, k vyřízení požadavku a vše končí zasláním kompletního uživatelského rozhraní s daty. Jednotlivé kroky jsou vzájemně synchronizovány [W8].

Naopak AJAX, díky XMLHttpRequest, může vyvolat libovolný počet nezávislých požadavků, jejichž výsledky mohou ovlivnit pouze patřičné části uživatelského rozhraní, bez nutnosti jeho celkového znovu načítání. Rozdíly ilustrují následující dva obrázky [W8].



Obr. 2 Synchronní & asynchronní model [W8]

4.6.3. Využití AJAXu

Velkým propagátorem AJAXu je zejména společnost Google, která na jeho základě vytvořila našeptávač Google Suggest, který podle prvních písmen nabízí ihned nejhledanější výrazy. Dále pak Google využívá AJAX u webového

rozhraní e-mailu GMail nebo Google Maps kde je touto technikou dynamicky načítán obrázek mapy. Stejnou techniku, pro načítání map, používá i Seznam a Atlas. AJAX je možné použít pro ulehčení výběru hodnot z několika závislých rozbalovacích seznamů. Po zvolení hodnoty prvního seznamu se do druhého seznamu načtou pouze položky související s položkou vybranou v prvním seznamu.



Obr. 3 Google Suggest – aplikace, která ajax zpopularizovala [W9]

4.6.4. Stinné stránky AJAXu

AJAX ke své funkci nezbytně potřebuje zapnutou podporu JavaScriptu. Lidí, jejichž prohlížeče tuto podmínku nespĺňují, přitom nemusí být málo - jedná se například o lidi, kteří prohlížejí internet pomocí vestavěného webového prohlížeče v mobilním telefonu nebo o lidi, kteří kvůli vyskakujícím reklamním oknům a zablokovaným pravým tlačítkům pro jistotu v prohlížeči JavaScript zakázali.

Dalším problémem je chování tlačítka zpět, protože to se používá jen pro statické stránky. Toto se dá bez váhání označit za největší problém technologie AJAX. Jak uvádí [W8] je tlačítko Zpět je hned po kliknutí na odkaz, nejčastěji používaným ovládacím prvkem prohlížeče. Při použití technologie AJAX však toto tlačítko vrátí uživatele na předcházející stránku, což ovšem neznamená návrat aplikace do předcházejícího stavu.

Při změnách na stránce pomocí technologie AJAX se nemění URL v adresním řádku prohlížeče. Proto není možné takto modifikovanou stránku poslat e-mailem nebo uložit do záložek.

5. API MAPOVÁ TECHNOLOGIE

Zpracováno dle Richarda Ondry [6].

API je zkratka slov application programming interface, což znamená rozhraní pro programování aplikací. Jde o sbírku procedur, funkcí či tříd nějaké knihovny, které může programátor využívat. API určuje, jakým způsobem se funkce knihovny mají volat ze zdrojového kódu programu. API jsou založena na známých standardních webových protokolech a mohou být rychle a snadno implementována v nejrůznějších jazycích, jako je PHP, JavaScript, Java...

5.1. Filozofie API

Důvod, proč velké firmy investují čas a peníze na pořízení a tvorbu dat, které pak zdarma poskytnou všem ostatním uživatelům je celkem praktický. Asi těžko by se podařilo těmto velkým firmám realizovat takové množství velmi specializovaných služeb, aby uspokojily potřeby všech uživatelů. Raději poskytnou data skrze API malým vývojářům, kterých je mnoho, a jejich nápady zůstávají nerealizované z důvodů nedostatku finančních zdrojů na pořízení vlastních dat. Tito se tak zdarma dostávají k zajímavým a nákladově náročným funkcím a datům. Z nich mohou stavět projekty, které by jinak byly zcela stranou zájmu vývojářů disponujících těmito zdroji.

Velké firmy nabízející API tak získávají zdarma reklamu a zvýšení povědomí o svých velkých projektech. Poskytovatelé také mohou sledovat aplikace vytvářené nad jejich API a zjišťovat, jaké funkce uživatelé postrádají, případně na ně zareagovat při vývoji aplikací. Což je cesta jak využít potenciál tisíce programátorů po celém světě ve prospěch firmy, aniž by jim musela něco platit.

5.2. Mashup – možnost využívat API a vytvářet nové služby

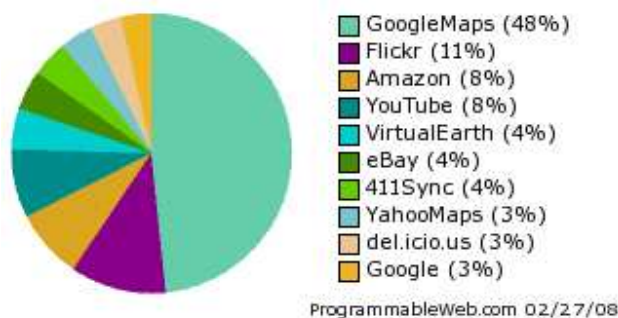
Mashupy (česky míchanice) jsou webové aplikace, které získávají data z jednoho nebo více zdrojů a tyto publikují jiným způsobem, v nových souvislostech. Vývojáři, mají možnost tato externí data využít prostřednictvím aplikačního rozhraní. Aplikace vytvořené na základě API jiných aplikací jsou často velmi silné a přitom pomoci minima programování a maximálního využití hotových API. Z kdysi složité

programovaných zakázkových aplikací se stávají hříčky pro jednotlivé uživatele.

Míchánice vycházejí ze dvou předpokladů:

- je vysoká poptávka po poměrně specializovaných službách, které se ale komplexně
- ve vlastní režii menšího vývojáře nevyplatí realizovat a pro větší není tato služba
- atraktivní,
- je vysoká nabídka široce použitelných řešení, které se nevyplatí realizovat pro
- jednotlivé možné specifické funkce v rámci velkých vývojářů. Přitom pro jejich
- praktické uplatnění je výhodné co největší rozšíření těchto řešení.

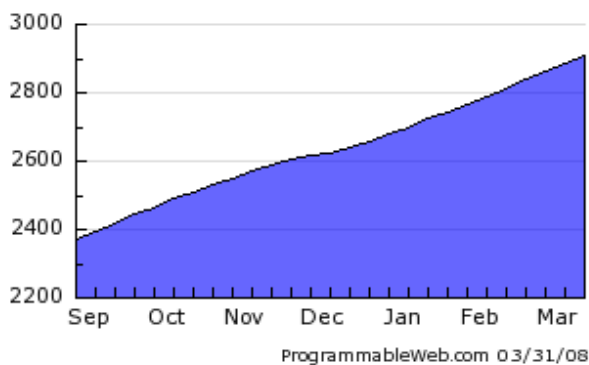
Tyto předpoklady daly základ vzniku Míchanicím, které mají většinou charakter ad hoc řešení pro konkrétní, úzce zaměřenou oblast, ale existují i univerzálnější projekty, často kombinující celou řadu zdrojů. Jedním z prvních zdrojů a inspirací pro míchánice se staly API map (zejména Google Maps, ale mapy takto poskytuje i Yahoo a u nás Seznam a Atlas). Oblíbenost mapových API ukazuje graf využití API pro Míchánice na obrázku 2.



Graf 1 Nejvyužívanější API pro Míchánice [13]

Velmi populární jsou i míchánice, které se snaží zachytit, co se děje na internetu. Princip spočívá v tom, že vezmeme jako zdroje agregátory novinek (nejčastěji RSS kanál - rodina XML formátů určených pro čtení novinek na webových stránkách) a dáme je dohromady, čímž vznikne „meta-meta-agregátor“. Příkladem kombinujícím až ohromující množství zdrojů je server Popurls – shromažďuje informace z více než 15 agregačních serverů (např. digg.com, reddit.com, Google News, Yahoo! News), fotky z flickru, videa z YouTube a dalších serverů. Oblíbené jsou i vyhledávače kombinující různé zdroje: např. MusicTonic při vyhledávání skupin a hudebníků dodá najednou fotky z flickru, videa z YouTube, alba, která si lze objednat z Amazonu, novinky z Yahoo [9].

Popularita míchanic v poslední době prudce vzrůstá, jak dokazuje i obrázek 3, znázorňující počet nových míchanic v adresáři webu Programmable za posledních 6 měsíců.

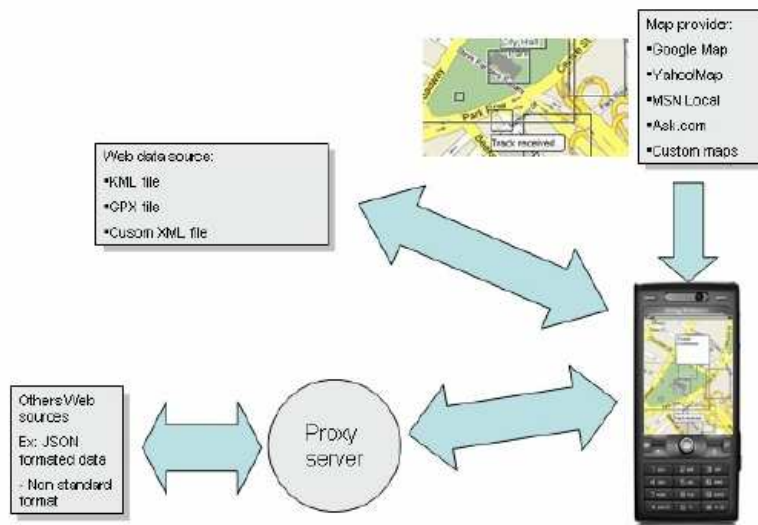


Graf 2 Mashup timeline [13]

Yahoo přišlo začátkem roku 2007 s projektem Yahoo Pipes. Jde o nástroj pro snadné vytváření mashup aplikací, který umožňuje v grafickém režimu jednoduše kombinovat zdroje (vyhledávání Yahoo, Yahoo local, Flickr, Google Base atd.), aplikovat na ně filtry a výsledek publikovat na svých stránkách. Každý si tak může vytvořit vlastní aplikaci, aniž by musel umět programovat – stačí pokročilé uživatelské znalosti.

Dalším kdo nabízí webové rozhraní pro vytváření míchanic je Google a jeho Google Mashup Editor, který však běží v testovacím provozu. Jeho rozhraní nám umožňuje vytvořit si vlastní míchanici ze služeb Google a vlastních dat. Vytvořená míchanice je pak zdarma hostovaná na stránkách Google, můžeme z ní také vytvořit Google Gadget, který si mohou uživatelé přidávat do personalizovaného iGoogle.

Mashup Editory umožňují i méně zkušeným zájemcům o tvorbu vlastních aplikací k vývoji takovýchto aplikací přistoupit. Je rozhodně snazší, používat mashup editory, než se učit PHP a snažit se zvládnout dostatek knihoven na to, aby aplikace nějak vypadala a fungovala. Mashup editory jen potvrzují vývojový trend míchání existujících zdrojů a nikoliv vynalézání již vynalezeného. Na druhou stranu je nutno říci, že pro běžné uživatele je tento koncept stále příliš složitý a slouží tak spíše jako hračka pro programátory [9].



Obr. 4 Mashup [W7]

6. API TUZEMSKÝCH A ZAHRANIČNÍCH POSKYTOVATELŮ

Tato kapitola je věnována prozkoumání základních služeb API nejvýznamnějších českých a zahraničních poskytovatelů.

Portály Seznam a Atlas byly vybrány, jelikož se jedná o jediné dva tuzemské poskytovatele, kteří nabízejí služby k vytváření mapových míchanic. Dalším poskytovatelem „obdobných“ služeb je zahraniční firma Google a jeho API pro Google Maps. Srovnání s tímto gigantem by však nebylo zcela na místě vůči českým portálům, jelikož ve světě internetu hraje čeština bezvýznamnou roli a tím pádem je menší i počet potencionálních uživatelů, kteří jsou hnacím motorem dalšího rozvoje těchto služeb. Pro násince však mají, tuzemské portály oproti světovému Googlu jednu velkou výhodu a to detailnější mapové podklady pro ČR [6] ...aktuální doplnění: Google v současnosti nabízí satelitní snímky o stejném rozlišení s možností většího digitálního zoomu oproti f. Seznam; topografická mapa je v sídelních detailech lepší u f. Seznam.

Uživatel má díky API k dispozici nástroj, umožňující doplnit vytvářené aplikace o prvek digitální mapy, který je k dispozici zdarma při dodržení jistých autorských podmínek. Vznikají tak aplikace, které lze charakterizovat jako mapové míchanice. Tyto mapové míchanice kombinují digitální mapy získané skrze API s tematickými informacemi z různých zdrojů. Stále více uživatelů objevuje možnosti digitálních map, jež je možné využít díky nabízeným rozhraním, aniž bychom se starali o samotné získání geografických dat a na internetu se tak objevují kvalitní mapové míchanice, které by v režii samotných poskytovatelů vznikaly jen stěží, jelikož oslovují malé skupiny uživatelům, kterým poskytují rozličnou specifickou funkcionalitu. [6]

Naproti tomu stojí vývojáři, kteří pracují na vývoji aplikace, jejíž služby poté prostřednictvím API nabízejí. Tito s rostoucího zájmu uživatelů jejich softwaru také profitují, jelikož na základě jejich připomínek a návrhů mohou poskytované rozhraní zdokonalovat a rozšiřovat o další funkce, které by byly uživateli hojně využívány. Tato skutečnost je však zřetelná spíše v zahraničí než na domácím poli, kde rychlejšímu rozvoji těchto služeb brání jednak omezené finanční zdroje pro podobné projekty, ale také nedostatečná komunikace mezi vývojáři a uživateli API, kdy vývojáři často přehlídí vhodné připomínky a návrhy na adresu jejich produktu. O tom svědčí i nízká aktivita na fórech zřízených k tomuto účelu, kde není výjimkou, že dotaz kladený přímo na vývojáře je ponechán mnohdy bez odezvy. Naprosto opačná situace je k vidění na fóru, jež je věnováno Google Maps API, kde žádné dotazy

nezůstávají bez odezvy a většinou prvními kdo na ně reagují, jsou právě vývojáři. Dalo by se říci, že samotní vývojáři si uživatele hýčkají a všemožně se snaží neztratit jejich přízeň, jelikož se naučily využít jejich potenciál ku vlastnímu profitu, což je skutečnost, která je na českém webu obecně opomíjena. [6]

6.1. České

6.1.1. Seznam

Pro užití mapových podkladů od společnosti Seznam je třeba souhlasit s podmínkami užití, které jsou oproti dále uvedeným podmínkách Atlasu podstatně striktnější. Zakazují použití map na jakémkoliv komerčním webu i v případě, že jde o bezplatné služby. Je zde také omezen počet požadavků na zobrazení hodnotou 1000 za den. Portál Seznam ve své podstatě jenom kopíruje prvky z API od firmy Google. Minimálně z prvních etap vývoje mapových služeb firmy Seznam by nezasvěcený člověk nevěřil, jak je možné, že se aplikace tak podobá svému zahraničnímu konkurentu – v dobách, kdy se vynakládá ohromné úsilí na boj proti kopírování myšlenek a nápadů. O to zarážející je potom fakt, že vedení f. Seznam uvalilo na používání tohoto produktu tak striktní a oproti f. Google velmi omezující podmínky.

6.1.2. Atlas

V případě, že chcete použít k tvorbě vlastní mapové aplikace službu API, která je poskytována portálem Atlas, musíte souhlasit s podmínkami užití. Podmínky užití nejsou nijak omezující v případě, že nechcete Amapy užít jako součást placené aplikace, tuto možnost Amapy nedovolují. [6]

6.1.3. Vzájemné porovnání

Zpracováno dle Richarda Ondry [6].



Obr. 5 Břeclav – Seznam [W10]

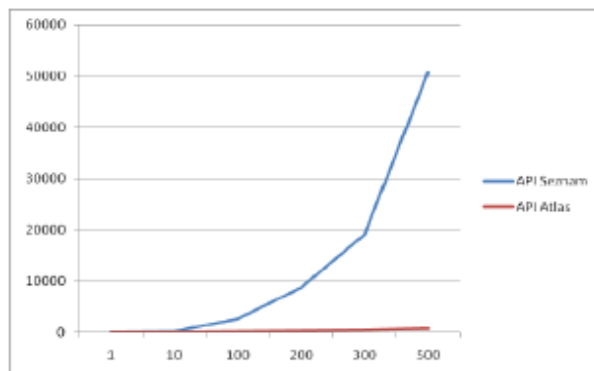


Obr. 6 Břeclav – Atlas [W11]

6.1.4. Rychlost načítání značek

Tab. 2 Rychlost načítání značek [6]

Počet značek	API Seznam(ms)	API Atlas (ms)
1	16	16
10	94	32
100	2563	172
200	8719	328
300	18968	484
500	50750	781



Graf 3 Rychlost načítání značek [6]

Autor tohoto porovnání neuvádí odkud byly dané body načítány, zda byly vepsány přímo do zdrojového kódu souboru, zda byly načítány z databáze, zda byly generovány nahodile-automatically, ..., v práci není zmínka o konfiguraci a hardwaru použitého PC. Nicméně pro ukázkou vzájemného porovnání to není důležité. Nesmíme se řídit výstupními časy, ale poměrným rozdílem mezi nimi. Výsledky nám jasně naznačují: křivka rychlosti načítání značek mapového portálu od f. Seznam roste exponenciálně v závislosti na počtu načítaných bodů, oproti tomu u f. Atlas se křivka od osy x odlepí opravdu velmi pozvolna.

6.1.5. Celkové zhodnocení (Seznam x Atlas)

Zpracováno dle Richarda Ondry [6].

Základní funkcionalita obou mapových aplikací nabízejících API je srovnatelná. Tím kdo je momentálně krůček napřed, co se rozšiřujících služeb týče, je Atlas, jehož rozhraní nabízí větší množství funkcí pro práci s externími datovými zdroji.

Tab. 3 Porovnání testovaných API [6]

	API Atlas	API Seznam
Klady	Podkladové a překryvné mapové vrstvy	Lepší satelitní fotografie a detailnější mapy měst a ulic ČR
	Široké možnosti doplňkových informací k bodům zájmu	Rychlejší načítání samotné mapy
	Podpora pro AJAX	
	Možnost vytváření fotogalerie pro body zájmu	
Zápory	Absence geokódování	Absence geokódování
		Maximální počet zobrazení denně 1000
		Chybí podpora souřadnicového systému JTSK

Je patrné, že portály přistupovaly k vývoji odlišným způsobem, z nichž každý má své klady i zápory. Zatímco Seznam se vydal cestou jednoduchého API, ve snaze přitáhnout co možná největší počet laiků (neprogramátorů), tak Atlas volil cestu propracovanějšího API, která bude zaměřena na uživatelsky zdatnější jedince. Zatím to vypadá, že lepší cestou se vydal Atlas, jelikož API je spíše hračkou pro více či méně zdatné programátory, kteří preferují rozsáhlejší funkcionalitu před jednoduchostí, nikoliv pro úplné laiky, kteří by ocenili onu jednoduchost. Této skutečnosti napovídá fakt, že většina mapových míchanic, jež u nás vznikly v posledních několika měsících, byla vytvořena nad rozhraním Atlasu nebo Googlu.

V obou API se nachází nedostatky, které sic nejsou rozsáhlé, často omezující. K jejich odstranění přispívají uživatelé, upozorněním v diskusním fóru kde však připomínky často zůstávají bez reakce vývojářů či zkušenějších uživatelů. Tato situace je obdobná v diskusních fórech k oběma mapovým aplikacím a přitom reakce od uživatele je jedním z klíčových artefaktů dobrého rozvoje API. Ve srovnání s Google Maps, kde denní počet příspěvků ve fóru se počítá v desítkách, jsou fóra věnovaná API portálů Seznam a Atlas žalostně prázdná a počet příspěvků je vhodnější počítat v týdenních intervalech a i tak málokdy přesáhnou desítku. Tento fakt, by mohl přispět k tomu, že většina uživatelů, kteří mají potenciál vytvářet kvalitní mapové aplikace, přejde od tuzemských poskytovatelů k API Googlu, kde se dočkají odpovědi na případný dotaz. Snad jsou si toho vědomi i vývojáři, kteří by se jako první měli snažit o to, aby na fóru věnovaném jejich produktu docházelo ke společnému řešení problému, jež by následně pomohly dalším uživatelům při tvorbě vlastní mapové aplikace a budou se tento problém snažit řešit.

Obě testované služby jsou stále ve fázi vývoje a jejich funkcionalita je doplňována a opravována, i když ne takovým tempem jaké by bylo ze strany uživatelů žádáno. Jejich momentální porovnání však vyznívá ve prospěch Atlasu, jehož rozhraní nabízí mnohem sofistikovanější metody, sice za cenu složitějšího API, ale zájem uživatelů jasně ukazuje tímto směrem. Čas ukáže, zda Seznam zareaguje na vývoj rozhraní Atlasu a doplní své API o rozšiřující funkce ve snaze přetáhnout uživatele na svoji stranu, nebo bude i nadále hrát druhé housle.

6.2. Zahraniční

Studie a zhodnocení zahraničních volně dostupných API mapových služeb.

Mezi zahraniční poskytovatele, kteří jsou v této kapitole vzájemně mezi sebou porovnávání patří: Google Maps, Yahoo! Maps a Virtual Earth (Microsoft).

Google maps API a Virtual Earth Interactive SDK (Microsoft) nabízí nejvíce funkcí, nejméně funkcí nabízí Yahoo!. Google nabízí nejvíce metod k poskytovaným funkcím.

Zobrazení mapy zadarmo na vlastních stránkách umožňují všichni tři poskytovatelé. Google a Yahoo! vyžadují zaregistrování na jejich stránkách a k ověření vyžadují ke každé aplikaci identifikátor. U Google je to API klíč a u Yahoo! se jedná o identifikátor API. Microsoft nic takového nevyžaduje.

6.2.1. Srovnání mapových dat pro Českou Republiku

Zpracováno dle Martiny Sochorové [7].

Microsoft a Yahoo! mají mapy podobně kartograficky zpracovány. Žádná z nich nezobrazuje budovy (u menších měst – cca 4000 a menší). Na mapách jsou vidět komunikace, železnice, vodní tok a Yahoo! má navíc ohraničený intravilán.

U Google je mapa co do podrobnosti na kvalitativnějši úrovni. Zobrazené komunikace jsou stejné třídy jako u Microsoftu a Yahoo!, ale mají podrobnější informace. Vodní tok a železnice jsou stejné, ale jsou zde navíc zobrazeny budovy.

Měřítko map nejsou sice stejná, ale i při přiblížení map od Microsoftu a Yahoo! Detailnější mapa k dispozici nebyla.



Obr. 7 Břeclav – Virtual Earth (Microsoft) [W12]



Obr. 8 Břeclav – Google Maps [W13]



Obr. 9 Břeclav – Yahoo! Maps [W14]

6.2.2. Celkové zhodnocení

Zpracováno dle Martiny Sochorové [7].

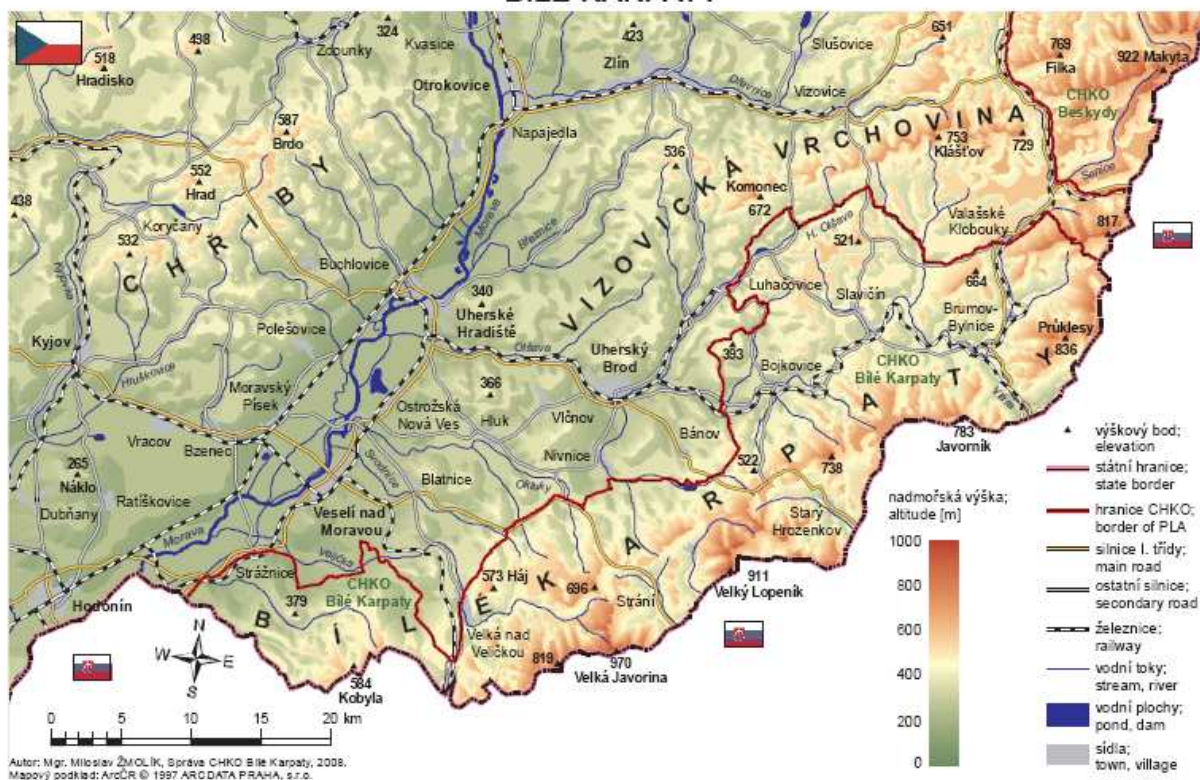
Yahoo! mapové API poskytuje nejméně funkcí. Po vlastní zkušenosti bych nedoporučila dělat složitější mashupy nad mapovým API od společnosti Yahoo!.

Pokud jde o zbývající poskytovatele, Microsoft Virtual Earth Interactive SDK se snaží držet krok s Google mapovým API. Vývoj API od společnosti google je však stále o něco napřed, poskytuje více komfortu a způsobů pro nastavení funkcí. Google má satelitní snímky o větším rozlišení než Microsoft (0,5m – stejné jako používá společnost Seznam na svém mapovém portále www.mapy.cz), ale Microsoft se snaží konkurovat pomocí leteckých snímků, které ale prozatím poskytuje jen pro několik významných měst.

Microsoft umožňuje nastavit národní prostředí na češtinu. Ovládací prvky nejsou k dispozici v češtině, ale informace zobrazené v informačním okně v češtině jsou, například u navigace.

7. CHARAKTERISTIKA ZÁJMOVÉHO ÚZEMÍ

CHRÁNĚNÁ KRAJINNÁ OBLAST (CHKO); PROTECTED LANDSCAPE AREA (PLA) BÍLÉ KARPATY



Obr. 10 CHKO Bílé Karpaty

Chráněná krajinná oblast Bílé Karpaty (dále jen CHKO BK) byla zřízena výnosem Ministerstva kultury ČSR č.j.17.644/80 ze dne 3.listopadu 1980, o zřízení chráněné krajinné oblasti, rozprostírající se na ploše 715 km² (skutečná výměra CHKO však dnes činí 746,6 km²) na území okresů Hodonín, Uherské Hradiště a Gottwaldov (dnes Zlín). Jejím řídicím orgánem je Správa CHKO.

Jedná se o bilaterální CHKO, kdy česká část má délku 70 km, orientaci severovýchod-jihozápad a leží v nadmořské výšce 175-970 m. Bílé Karpaty představují mimořádnou oblast mezi našimi velkoplošnými chráněnými územími především proto, že jsou nejvyšším pohořím jihozápadního okraje vlastního karpatského horského systému. Celá oblast, ale zejména její jižní část, byla po mnoho staletí kultivována člověkem. Přesto, nebo právě proto se zde dochovaly mimořádně cenné přírodní hodnoty a na mnoha místech lze hovořit o harmonické krajině. Pro tyto přírodní a krajinné kvality byly Bílé Karpaty v rámci programu Člověk a biosféra (MAB) organizace UNESCO dne 15.4. 1996 zařazeny mezi evropské biosférické rezervace. Význam tohoto

území dokazuje i udělení Evropského diplomu pro chráněná území v roce 2000.

Svým charakterem mohou Bílé Karpaty sloužit jako modelové území pro koexistenci zájmů ochrany přírody s hospodářskými aktivitami respektujícími ekologickou únosnost území a jeho přírodní podmínky.



Obr. 11 Krajina CHKO Bílé Karpaty

Rozsáhlá historická odlesnění v Bílých Karpatech měla velmi často charakter krajinářských úprav citlivě využívajících zdejších přírodních podmínek. Výsledkem jsou tisíce hektarů jedinečných květnatých luk s roztroušenými dřevinami, představující dnes typický krajinný ráz Bílých Karpat. Z přírodovědného hlediska jsou tyto květnaté karpatské louky pozoruhodné především bohatostí rostlinných společenstev s vysokým zastoupením kriticky ohrožených druhů rostlin. Díky tomu patří k nejcennějším lučním biotopům Evropy a jsou studijní plochou světového významu. Dalším neméně cenným prvkem jsou rozsáhlé lesní komplexy v centrální a severní části pohoří z celou řadou typických prvků karpatské květeny i fauny.

Krajinný ráz střední a severní části Bílých Karpat je dotvářen poměrně řídkým osídlením pasekářského či kopaničářského typu, absencí velkých průmyslových podniků a zachovalou architekturou celých obcí (Lopeník, Vyškovec, Žitková). Pro západní část CHKO jsou charakteristické velmi rozsáhlé komplexy květnatých luk s rozptýlenými soliterními stromy. Střední část CHKO v širším okolí Starého Hrozenkova se nazývá Moravské Kopanice. Její současný vzhled vznikl teprve velmi pozdní valašskou kolonizací v 17. - 18. století a vyznačuje se roztroušenou zástavbou, střídáním zalesněných a bezlesých ploch s mozaikou sušších míst, mokřadů, drobných lesíků, křovin a nevelkých políček. Severovýchodní část pohoří v okolí Valašských Klobouk a Brumova patří k Valašsku. Krajina zde již připomíná Javorníky, které na Bílé Karpaty bezprostředně navazují.

Rozmanité způsoby hospodaření, různorodý historický vývoj a v neposlední řadě odlehlost od průmyslových středisek umožnily zachovat neobvykle vysokou biodiverzitu na mnoha typech stanovišť, od teplomilných šipákových

doubrav po pralesovité horské bučiny, od teplomilných stepních porostů k podhorským přepásaným loukám a nejrůznějším typům drobných lesních i lučních mokřadů. Bílé Karpaty se staly pojmem především jako území s nejvyšší diverzitou a s největší kvantitou vstavačovitých rostlin (orchidejí) ve střední Evropě.

Přírodní i kulturní faktory tak vytvářejí z Bílých Karpat území mimořádně cenné i v evropském kontextu.

[W17]

8. GOOGLE MAPS API

Postupy pro vytvoření aplikace založené na mapových podkladech volně dostupných skrze rozhraní API jsou u většiny poskytovatelů obdobné. V následující části textu budou popsány kroky, jak docílit umístění Google Maps do vlastní webové stránky uživatele.

8.1. API klíč

K tomu, aby výsledná aplikace fungovala na uživatelských stránkách je zapotřebí získat mapový API klíč (Google Maps API Key). O tento klíč lze požádat na adrese <http://code.google.com/apis/maps/signup.html>.

Podmínkami k získání tohoto klíče je

- zaregistrování v doméně gmail.com,
- vlastní webová stránka a
- souhlas s licenčními podmínkami.

Jednotlivý mapový API klíč je generován pro unikátní název webové stránky a jeho kořeny [W33][7].

8.2. Licenční podmínky

Níže jsou uvedeny některé hlavní body licenčních podmínek [W33]:

- Neexistuje limit na množství zobrazení stránek, které lze vygenerovat za den s využitím mapového API.
- Limit žádostí o geokód za den je 50 000 mapových API klíčů.
- Mapové API nesmí obsahovat reklamu.
- Výsledná aplikace musí být volně přístupná.
- Nesmí být měněna nebo zastíněna loga nebo názvy na mapě.
- Google periodicky aktualizuje API a pro aktualizaci příslušného místa (stanoviště) musí být použita nová verze API.
- API nesmí být použito například k identifikaci místa pro nakupování ilegálních drog ve městě nebo nějaký podobný ilegální aktivit, k identifikaci soukromých informací o soukromých jedincích.

Plné znění licenčních podmínek je na adrese [W33][7].

8.3. Dokumentace Google API

Dokumentace k mapovému API Google se nachází na stránce

[W33]. Tato stránka obsahuje kromě souhrnného výčtu všech funkcí, které nějakým způsobem s mapovým podkladem pracují, také mnoho konkrétních praktických ukázek (mashupů).

8.4. Základní mapové objekty

Asi nemá smysl přepisovat a jednotlivě popisovat funkce ze stránek společnosti Google, které s mapovým podkladem pracují. [W34] Nicméně pro informaci jsou níže uvedeny nejzákladnější a nejvyužívanější objekty, na kterých je aplikace „Inventarizace ovocných sadů a stromů“ postavena. Vypsání objektů má mnoho vnořených metod, které jsou vývojáři ze společnosti Google nadále průběžně rozšiřovány.

GMap2 – objekty této třídy definují samotnou mapu na stránce a veškeré její náležitosti (typ mapy, střed mapy, ovládací prvky, atd.)

GInfoWindow – informační okno („bublina“)

GPoint – bod v mapě dle pixelového souřadného systému (ne WGS84)

GMarker – bod v mapě dle souřadného systému WGS84

GPolyline – polylinie

GPolygon – polygon

GIcon – objekt pro definici ikon umístěných v mapě

GMarkerManager – sada funkcí pro správu bodu

GOverlay – instance vkládající objekty do mapy

...

8.5. Úskalí aplikace

Rychlost celého systému je závislá na mnoha faktorech. Mezi nejdůležitější z nich patří:

- rychlosti připojení k internetu
- rychlosti databázového serveru
- RAM paměť PC na kterém aplikace pracuje
- použitý prohlížeč

Ve chvíli, kdy počítač je méně výkonné konfigurace dohromady s pomalejším přístupem k internetu, je velmi pravděpodobné, že se prohlížeč při například načítání jablek zasekne. Načítání záznamů je mnohdy opravdu velmi náročnou záležitostí a ne vždy je možné načíst všechny záznamy zároveň.

Z tohoto důvodu byly vytvořeny v systému tzv. kvadranty (názorná ukázka – viz. **MANUÁL** k aplikaci – „**Uživatelská funkcionality**“ - „**Kvadrant**“).

Jejich využití spočívá v rozčlenění zájmového území (CHKO BK) na 4 dílčí jednotky. Načítání záznamů pak nemusí probíhat na celém území (tj. nemusí se z databáze načítat všechny objekty), nýbrž postačí pouze vybrat požadovanou podoblast do které se načtou objekty do ní spadající (Obr. 22 Přidání externí vrstvy).

Není možno konkrétně popisovat co PC zvládne načíst a co již ne. Faktorů, které rozhodují o dynamice aplikace je mnoho. Uvedu snad jen jeden konkrétní příklad, kdy již aplikace přestane pracovat.

Př.

- PC Intel Pentium M, 512MB RAM, 2MB L2 cache, 128MB VRAM (grafická karta)
- připojení k internetu WI-FI 1MB
- databáze uložena na serveru www.wz.cz

Výsledek: při načítání v prohlížeči IE aplikace nezvládne načíst více jak 400 bodů

8.6. Nejzajímavější aplikace vytvořené nad mapovými API (nejen Google)

Zpracováno dle Martiny Sochorové [7].

Oblasti využití API Mapových služeb

- Běžné události
- Doprava
- Počasí a Země
- Pivo & Víno
- Bydlení & Nemovitosti
- Obchod
- Zaměstnání
- Statistika & Demografie
- Historické mapy
- Rekrece & Fitnes
- Přeprava & Turismus
- Významné osobnosti
- Knihy
- Námořní mapy

London Executive

Aplikace vytvořená nad Virtual Earth Interactive SDK se týká prodeje a pronájmu nemovitostí v Londýně.

V levé části je panel se čtyřmi záložkami. První obsahuje nastavení, jaké byty mají být zobrazeny (v jaké cenové relaci, s kolika pokoji, zda pronájem nebo prodej bytu). Druhá záložka obsahuje popis nemovitosti, fotografie a odkaz na internetové stránky kde je nemovitost detailněji popsána. Třetí záložka obsahuje vyhledávání pomocí adresy. A čtvrtá záložka je help k ovládání této aplikace.

V pravé části je panel s nástroji k ovládání mapy (změna měřítka, posun mapy, změna typu mapy). Přiblížení a oddálení se dá ovládat pomocí kolečka na myši, ale nesmí být zapnut typ mapy "Bird's eye".



Obr. 12 Mashup [W23]

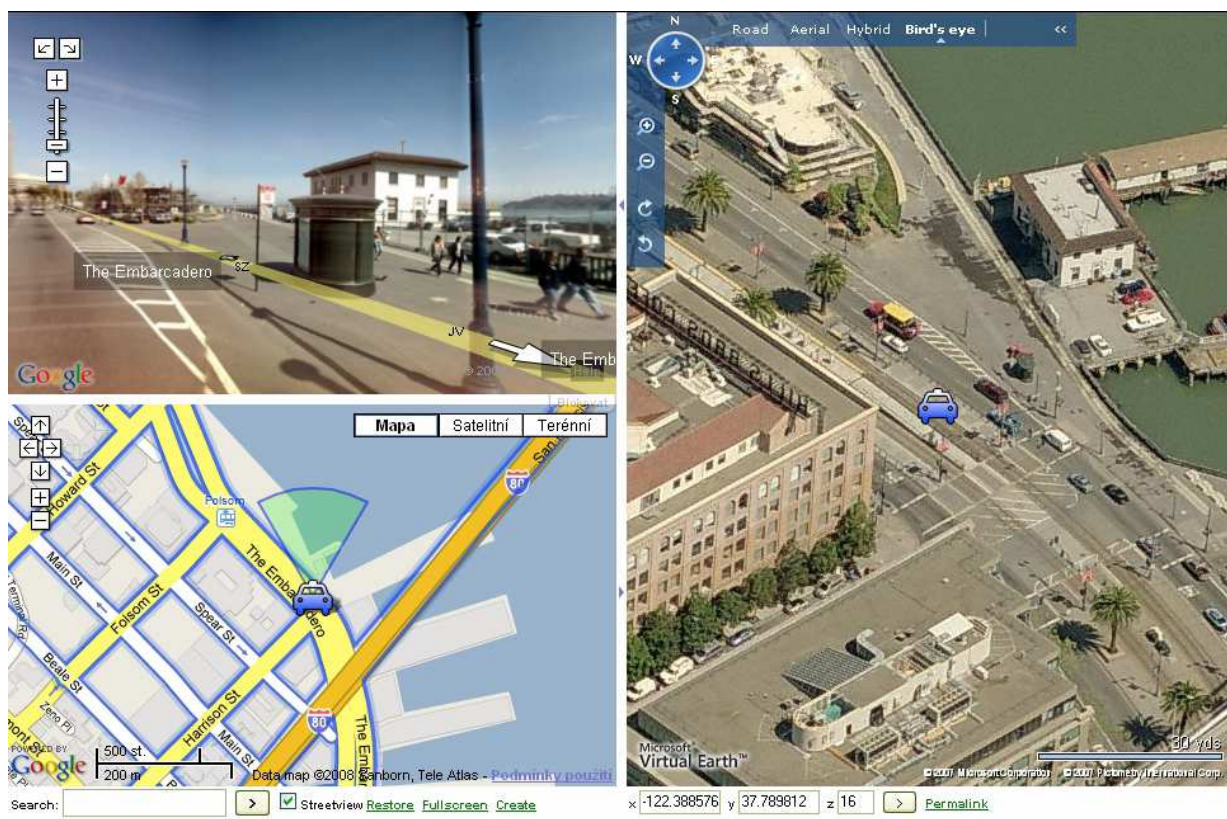
Dual Maps

Dual maps umožňuje zobrazit mapy poskytovatelů Google a Microsoft vedle sebe a s propojenou funkcí. Dual maps také zahrnuje nové Google Street View. Středové „dělítko“ umožňuje maximalizovat Google mapu včetně Street View. Volitelně lze na mapě zobrazit ukazatele. Nastavení lze přizpůsobit pro vlastní potřeby. Standardní nastavení zobrazuje typ mapy Google "road" na levé straně a Virtual Earth "Bird's eye" na pravé straně.

"Bird's eye" není dostupné pro všechny oblasti. Jestliže "Bird's eye" je nedostupný, je použit typ mapy „hybrid“ [W26].

Po výběru typu mapy "Bird's eye" jsou dostupné další parametry pro nastavení směru jízdy a středu mapy. Toho může být využito k přizpůsobení pohledu z jednotlivých míst a je užitečné pro zobrazení obrázků velkých budov nebo staveb. Standardně je nastaven pohled na San Francisco a ukazuje kombinaci Street View, mapy Google a Virtual Earth "Bird's eye" [W26].

Dual Maps jsou volně vloženou funkcí na uživatelské webové stránce a nevyžaduje registraci k použití [W26].

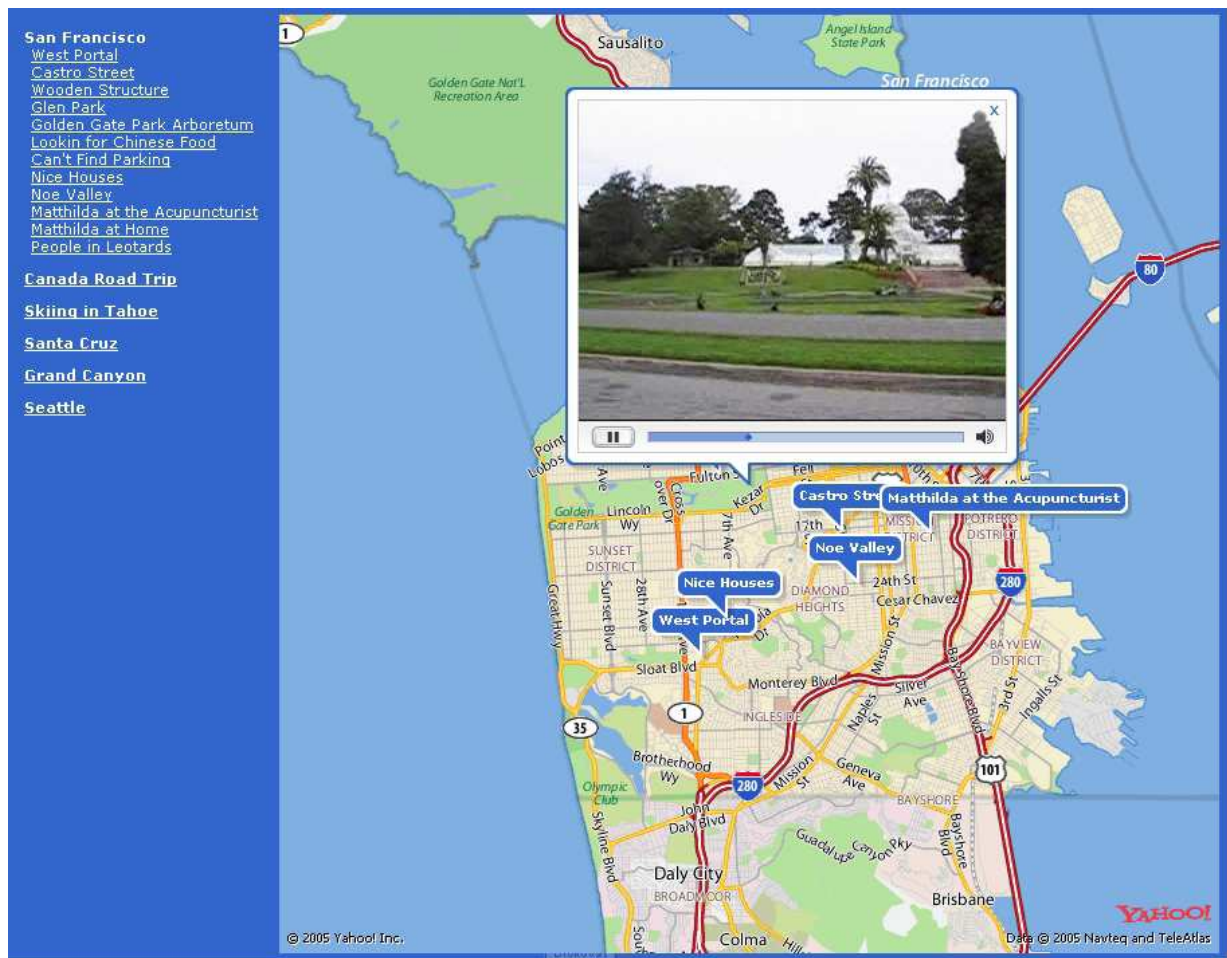


Obr. 13 Dual Maps [W24]

Zaznamenání života v San Francisku pomocí náhodných videí

Způsob života v San Francisku je zaznamenán na krátkých videozáznamech z různých čtvrtí okolo města, vytvořených Derekem Lu [W25].

Aplikace je složena z legendy v levé části a mapy v části pravé. Legenda obsahuje názvy lokalit, k nimž jsou videa natočena. Video lze spustit buď kliknutím na informační okno nebo na odkaz umístěný v legendě.



Obr. 14 Život v San Francisku [W25]

9. METODY A POSTUP ZPRACOVÁNÍ

9.1. Výběr prostředí

Nároky na aplikaci uvedené v zadání tohoto projektu:

- free webové řešení
- přístup možný odkudkoli, kde je připojení k internetu
- spuštění bez instalace do systému

Na základě těchto požadavků bylo nutno vybrat nejvhodnější řešení. Nejreálněji se jevily dvě možnosti. Jedna cesta vedla přes využití libovolného mapserveru (Minnesota MS, apod.), druhá skrze využití API mapových portálů. Do obou řešení by se daly vložit a poté následně editovat všechny záznamy.

Mapserver je silný nástroj. Jeho opensourcová zpracovanost, kdy je pro tvůrce předpřipraveno mnoho funkcí a grafických šablon, by jistě splňovala veškeré požadavky na tento projekt kladené. Další z kladů především mnou vyzdvihovaných je fakt, že v mém blízkém okolí vznikla řada projektů založených na tomto řešení. Případné problémy by byly tudíž možno konzultovat právě s těmito autory. Krom výhod to má však i své nevýhody a to především v porovnání s řešením druhým, kterým jsem se nakonec vydal.

Data. Problém dat zmiňovaným již v kapitole 10.3. Data. Mapserver, ač nástroj splňující veškeré na projekt kladené požadavky, stále vyžaduje vstupní data na základě kterých by se digitalizace prováděla. Avšak sehnání (podrobné, názorné, přehledné) vstupních vrstev pro území celého CHKO Bílých Karpat a do budoucna i mimo toto území (tj. nutnost přidávání) není jednoduché.

Zvolil jsem řešení jehož existence je poměrně nová, nicméně pro účel této aplikace (digitalizace) vhodnější. Základní myšlenkou mapových API je poskytování stále aktualizovaných (dle možností společnosti) mapových rastrových podkladových vrstev. V souladu s dnes již poměrně vyspělou nabízenou funkcionalitou ovládání těchto vrstev, obecně se zrychlující konektivitou uživatelů k internetu se většina PRO přiklání k tomuto řešení.

9.2. Inspirace

V zadání diplomové práce byla popsána základní funkcionalita výsledné aplikace. Nadefinovány však byly pouze hlavní pilíře. Způsob jakým se bude s aplikací pracovat, její barevné sladění, intuitivnost při ovládání a mnoho

dalších, to vše již spočívalo na mém subjektivním rozhodnutí. Inspirací k tomu mi bylo nepřeberné množství internetových stránek, které jsem kvůli tomuto účelu prošel.

Vstupní branou do problematiky jsou oficiální stránky Google Maps API [W2], kde kromě popisu všech funkcí se nalézají i krátké příklady využití. Dále pak internetové a hojně navštěvované diskuse spravované zaměstnanci z vývoje Google Maps API – americké: [W15] a jejich český ekvivalent (mnohem méně obsáhlý): [W16]. Jedním z dalších kvalitních zdrojů jsou stránky Mika Williamse [W3] s velkým množstvím praktických ukázek využití Google Maps.

Alespoň ze začátku mi byl prospěšným i zdroj tištěný. Kniha „Google maps hacks“ [1] popisující základní triky a tipy práce s Google maps a kniha Beginning Google Maps Applications with Rails and Ajax [2] věnující se problematice API již s ohledem na využití AJAXu.

Ač se na internetu nachází nepřeberné množství různorodých aplikací založených na mapových API, jen malé množství z nich jsou složitějšího charakteru. Většinou jejich funkcionalita končí natažením základní mapy do webových stránek a nad ní umístění ikon s popisem.

Programování je jako lego, můžete „vymodelovat“ téměř cokoli a jakkoli. Záleží pouze na vaší fantazii a následné programátorské schopnosti převést myšlenky do zdrojového kódu.

9.3. Data a jejich konverze

Data z probíhající inventarizace v letech 2001 – 2005 na území CHKO Bílých Karpat mi byla poskytnuta v podobě dvou shapefilů. Jeden tvořil bodovou vrstvu reprezentující ovocné stromy, druhý vrstvu polygonovou znázorňující ovocné sady. Data pro účely použití v prostředí Google Maps musela být konvertována, u polygonové vrstvy navíc i šifrována do požadované podoby.

9.4. Aplikace

Práce probíhala modulárním způsobem. V hlavě „načrtnutou“ finální aplikaci jsem si rozdělil do mnoha dílčích kroků, např. vytvoření bodu, ukládací procedury, přidávání vrstev do projektu, výběry z databází, šifrování... Ve chvíli zvládnutí všech dílčích kroků jsem začal celý systém skládat dohromady.

9.5. Sepsání dokumentace

Posledním krokem je sepsání vysvětlujícího textu, jak vlastně aplikace vznikala, potřebné znalosti, využití softwarové prostředky, apod. Součástí této ucelené podoby je i vlastní návod, který popisuje základní práci s aplikací - MANUÁL.

10. POSTUP PRÁCE

10.1. Výběr API mapového prostředí

Zamyšlení při výběru: API mapové technologie jsou ve webovém prostředí poměrně novou záležitostí. Společnost Google začala svůj API vývoj kolem roku 2006. Aby konkurenční společnosti nezaspali tento nepokrytý úsek trhu, během krátké doby začali (většinou známé zahraniční i české vyhledávací servery – Yahoo, Seznam, ...) vytvářet obdobné systémy. Letos se píše rok 2009 a řekl bych, že na trhu je v současnosti dostatečné množství silných hráčů, kteří zamezí vzniku nových. Společnosti, většinou zahraniční, jsou již tak dobře na trhu etablované, že málokdo je schopen přijít v této oblasti s něčím takovým, co by dokázalo konkurovat tomuto rozjetému vlaku... Snad nejsilnějším hráčem (a nejen v této oblasti) se stala společnost Google, která krom nesporného kapitálu jak finančního, tak v lidských zdrojích, operuje z velké části na opensourcové filosofii. Krom jiného, tj. zadarmo. Pokud firma zvládne tuto filosofii dobře zviditelnit, pak se stává v konkurenčním boji téměř nepřekonatelná.

V době, kdy jsem se rozhodoval nad mapovým prostředím, pod kterým budu aplikaci vyvíjet, ještě zdaleka nebylo tolik možností na výběr, jako je tomu dnes. Nabízená funkcionalita prostředí zvládala mnohdy pouze základní operace.

Z tehdejších domácích poskytovatelů existovala pouze společnost Seznam. Ze zahraničních byli nejznámější Google Maps a Yahoo! Maps... V době vzniku této aplikace (polovina roku 2007) jsem měl o API povrchní znalosti. Po přečtení pár článků o této problematice jsem se spíše intuitivně rozhodl pro zahraniční Google. A že jsem neudělal vůbec špatně si nyní, kdy jsem již ve znalostech z oblasti API na daleko vyšší úrovni, ověřuji nejen z vlastní zkušenosti, ale také z rozrůstajících se odborných diskuzí na toto téma. Český Seznam v té době byl opravdu v „plenkách“ a hlavně měl tehdy velké množství omezení, které mě odradily. Je sice pravdou, že rozlišení a kvalita jím poskytovaných snímků ČR byla na vyšší úrovni (v současnosti již Google používá satelitní snímky stejného rozlišení, topografická část je u Seznamu stále kvalitnější), ale pořád to byla jen jakási česká kopie zahraničního kolegy. Google byl a je na trhu největší hráč, už tehdy jeho formát KML konsorcium OGC ohlásilo jako oficiální. Ať člověk má rád společnost Google, nebo nemá, jeho vzrůstající vliv nejen (opravdu nejen) na poli GIS stoupá nezadržitelně vzhůru. V tomto má ekonomika jasná pravidla: peníze a vývoj se v největší míře udržují tam, kde je uživatelů nejvíce.

V neposlední řadě bych se rád zmínil o velmi otevřené politice společnosti Google ke svým produktům. Firma se snaží vycházet vstříc inovativním vývojářům tím, že jim umožňuje dotvářet většinu svých vlastních produktů. Ať již se jedná o poskytování map, softwaru pro mobilní telefony či zdrojového kódu pro nový prohlížeč Chrome, aj. A že je tato politika správná ukazují finanční výsledky společnosti. V dobách, kdy se rozjíždí ekonomická krize (2/2009) po celém světě, dynamika ekonomických výsledků firmy Google je poměřitelná jen s málokterou současná organizací. Víím, můžeme namítat, že peníze nejsou zárukou kvalitního systému (např. věčný spor OS Microsoft Windows a UNIX). Souhlasím. Nicméně společnosti Google se to prozatím velmi dobře daří vyvracet.

Uvědoměním těchto nesporných faktů jsem se rozhodl vyvinout aplikaci „Inventarizace ovocných sadů“ na bázi Google Maps. Věřím, že tímto výběrem jsem si zajistil jakousi její dlouholetost. Do vytvoření aplikace jsem vložil nemalé úsilí a nerad bych, aby celá tato práce vzala za své jenom kvůli tomu, že se poskytovatel těchto služeb neudržel na trhu. Nyní již s více jak ročním odstupem víím, že jsem se rozhodl správně.

10.2. Data

Vstupní data ovocných stromů a sadů z inventarizace probíhající na území CHKO Bílých Karpat v letech 2001 až 2005 mi byla poskytnuta v SHP formátu. Jednalo se o dvě vrstvy – polygonová (sady – počet 27) a bodová (stromy – počet 845). Převod z SHP do Google Maps prostředí ač se zdál ze začátku docela jednoduchý, se záhy značně zkomplikoval.

Poskytnutá data byla v zobrazení S-JTSK. Google Maps pracují v zobrazení WGS-84.

Bodovou vrstvou (stromy) jsem v ArcMapu převedl použitím fce „define projection“ z S-JTSK do požadovaného WGS-84. Nyní jsem aplikoval externí skript (psaný v Avenou), kterým jsem získal z SHP souboru ze sloupce ID hodnoty geografických souřadnic X a Y. Nyní již následoval krok „překlopení“ těchto dat do databáze MySQL. Bylo potřeba vytvořit z DBF souboru soubor SQL. K tomu jsem použil programy MS Excel a následně JanDat, který je součástí free systému Janitor. Následný export do phpMyAdmin již byl díky jeho intuitivnímu prostředí jednoduchou záležitostí.

Polygonová vrstva (sady) a její postup převodu byl o poznání složitější. Zde jsem narazil na problematiku ukládání polygonu z obecného hlediska. Každý polygon je tvořen lomovými body – vertexi. Každý vertex je bod o vlastních souřadnicích X a Y. Pakliže vezmeme v potaz např. polygon o 100 vertexech,

rázem máme 200 float čísel k uložení. A to v případě mnou vyvíjené aplikace uložení ne na rychlý vlastní harddisk počítače, ale přes internetové spojení do databáze. Zde si dále musíme uvědomit způsob uložení – každý bod na jeden řádek databáze(?), jako posloupnost čísel oddělených např. středníkem(?)? Obdobných řešení je mnoho, ale ve své podstatě většina jsou svým způsobem velmi nesofistikovaná a náročná na ukládání.

Google pro tento problém použil šifrovací algoritmus (Douglas-Peucker). Jeho vysvětlení je na adrese [W18]. Jedná se v podstatě o způsob, kdy se hodnota např. souřadnice X zašifruje do řetězce ASCII znaků. Stejně je tomu i u Y hodnoty. Po vykonání těchto kroků se přechází k bodům dalším, kde se zpracovávají (zašifrovávají) obdobným způsobem. V konečné fázi se všechny jednotlivé řetězce sloučí do řetězce celkového, s kterým je již operace ukládání daleko snazší. Tzn. do databáze se nyní ukládá ne 200 čísel, ale např. řetězec o 600-ti znacích.

Z kartografického hlediska významnou výhodou šifrovacího způsobu je také následná možnost použití generalizace (generalizace bodů - viz. 10.6.1. Generalizace). Bod si po zašifrování pamatuje své přesné souřadnice a vzdálenost ke svým sousedům. Z tohoto se následně vytváří další řetězec znaků, který nám říká při jaké úrovni přiblížení se daný bod zobrazí a kdy zůstane skrytý. Vysvětlení tohoto algoritmu [W19]. Konkrétní a pěkný příklad funkčnosti šifrovacího algoritmu [W20] viz. Obr. 15 Kódovací utilita.

Center map at:

Mapa Satelitní Hybridní

Latitude:

Longitude:

Min zoom level:

Point list

(37.44801517458876,-122.15578079223633) Lev	▲
(37.451558445682544,-122.15492248535156) Le	
(37.455101548940156,-122.15269088745117) Le	
(37.455919164318125,-122.14994430541992) Le	
(37.45564662685195,-122.14719772338867) Lev	
(37.45632796865522,-122.14462280273437) Lev	
(37.45523781879053,-122.14118957519531) Lev	
(37.45373883678061,-122.13912963867187) Lev	
(37.4530574713902,-122.13552474975586) Leve	
(37.452921197567285,-122.13277816772451) Le	
(37.45278492349606,-122.12968826293945) Lev	
(37.45278492349606,-122.12968826293945) Lev	
(37.45278492349606,-122.12968826293945) Lev	
(37.45278492349606,-122.12968826293945) Lev	▼

Encoded Polyline:

Encoded Levels:

Obr. 15 kódovací utilita

V ArcMapu obdobně jako tomu bylo u bodové vrstvy (stromy) se změnilo zobrazení z S-JTSK na WGS-84. Jednotlivé polygony (sady) jsem následně v prostředí ArcView 3.1 pomocí skriptu převedl na vrstvu bodů (vertexů). Použitím dalšího skriptu na tuto vrstvu se vyexportovaly ze sloupce ID hodnoty souřadnic již ve formátu WGS-84. Autor Mark McClure zabývající se touto problematikou vytvořil na svých stránkách [W4] užitečné aplikace. Pomocí jedné z nich jsem docílil převodu vertexů právě do požadované zakódované podoby. V této části byl postup opět obdobný, jako u vrstvy stromů. Tj. soubor DBF se využitím programů MS Excel a JanDat uložil jako SQL. Následný import (27 záznamů – polygonů) do phpMyAdmin a ruční vložení zašifrované podoby polylinie ke každému z nich.

10.3. Vytvoření aplikace

Klíčovou částí celé práce bylo naprogramování systému, který dokáže uchovávat, editovat a následně vizualizovat objekty na mapovém podkladu poskytnutém společností Google.

Přístup k mapovému podkladu je zprostředkováván prostřednictvím skriptovacího jazyka JavaScript. Skrze něj jsme schopni ovládat kompletně prostředí mapy, měřítko, načítání bodů, přibližovací a oddalovací fce, aj. Vše běží na webové stránce naprogramované v HTML za použití kaskádových stylů CSS. Jako jakýsi prostředník mezi mapovým oknem a databází slouží programovací jazyk PHP. K samotné práci s jednotlivými záznamy v databázi je využíván jazyk MySQL.

10.3.1. Použité nástroje:

- www.wz.cz - freehosting
 - phpMyAdmin 2.6.0-pl3
 - PHP 4.3.4
- JavaScript
- HTML
- CSS
- MySQL

Aplikace byla vyvíjena na free web hostingu. Byl využit prostor na serveru www.webzdarma.cz, jenž ač přes svou mnohdy nespolehlivost a pomalý přístup, svému účelu posloužil. Tento prostor i se svými službami je poskytován zdarma, proto si myslím, že není na místě jeho velká kritika. Co se týče např. rychlosti stahování záznamů z databáze, tak se s komerčními prostory nemůže rovnat.

10.3.2. Filosofie aplikace

Aplikace jak již bylo v úvodu naznačeno vznikala modulárním systémem. Tj. na začátku byla práce kompletně navržena na papír. Bylo stanoveno jak má aplikace pracovat, jak se chovat, jak má být ovládána. Na základě tohoto konceptu byl tento celek rozdělen na množství jednotlivých částí, které se následně postupně technicky řešily do takové podoby, aby ve finále šly veškeré části seskládat do jednodušší aplikace.

Ač se mnohé řešení konkrétních dílců jevílo triviálně (kór, když na internetu kolují spousty obdobných a vypracovaných projektů) řešení nebylo mnohdy zas až tak jednoduché. Je pravdou, že inspirace a prostudování různých návodů mi mnohdy problémy vyřešily (i když samotné vyhledání vhodného řešení na internetu je mnohdy záležitost opravdu spousty hodin).

Pro dokončení této práce bylo pro mne velmi důležité, nicméně o to více zdouhavé samotné pochopení jednotlivých programovacích jazyků a jejich vzájemné propojení. V programátorské „branži“ se nestává tak často, aby se v jedné aplikaci spolupracovalo tolik jazyků... Který z nich má přednost, který jazyk vstupuje do procesu jako první, kdo vytváří hodnoty pro toho druhého, apod. Práce alespoň pro mne nebyla díky množství jazyků, které musely

pracovat naprosto v souladu vůbec ze začátku jednoduchá (ač jsem se s většinou těchto jazyků začínal seznamovat teprve při tvorbě aplikace, tento názor sdílejí i daleko zkušenější programátoři). „Narovinu mohu říci, že jsem ze začátku nechápal, jak to vlastně složit celé dohromady.“ Byly chvíle, kdy aplikace pracovala tak jak dle mého měla, zatímco jindy se chovala naprosto mimo mou logiku.

Nechci tvrdit, že finální podoba aplikace je nějaké veledílo, ale aby člověk sestrojil něco podobného, musí již něco o této problematice „tušit“. Zní to opravdu velmi prostě, ale člověk to musí nejprve celé pochopit.

Zkusím zde vypsát nejzákladnější potřebné programátorské znalosti pro vytvoření aplikace:

JavaScript, API:

- bod – vytvoření; změna ikony; změna polohy; smazání; skrytí; tisk bodu; zvětšení klikací oblasti; stíny; animace; aj.
- polygon – vytvoření; editace; smazání; skrytí; změna barvy; kódování; generalizace; aj.
- mapa – navedení vhodných ovládacích entit; zoom funkce; překryvy mapy georeferencovanými vrstvami; posluchače a vyvolání přerušení při kliku myši; aj.
- reload a přesměrování stránky; ovládání objektů mapy; práce s řetězci; tisk; navádění externích skriptů; aj.

PHP – transfer dat z databáze do HTML stránky a naopak; zajišťování správné češtiny; kontrola oprávněnosti vstupu; escapování znaků; aj.

MySQL – zápis; editace; export; import; znaková sada; aj.

CSS – komplexní grafické návrhy stylů

HTML – komplexní práce při vytváření webové stránky

Grafické programy – tvorba ikon a úprava dekoračních obrázků

Detailní popis jednotlivých částí aplikace nemá význam – minimálně by značně přesáhl rozsah této diplomové práce. Jejich funkční využití je popsáno v kapitole 10.6. Co aplikace obsahuje a jak pracuje.



Obr. 16 Model aplikace

10.4. Testování aplikace


Samotnému spuštění aplikace do provozu předchází více částí. Jedna z významných logicky spočívá v naprogramování celého systému do podoby, v níž by neměly nastat neočekávané situace. K naplnění tohoto cíle je nutno již během celé etapy vývoje tento software testovat. Častým problémem při vytváření webového produktu je jeho vzájemná kompatibilita s internetovými prohlížeči.

Aplikaci jsem vytvářel a zároveň testoval na PC, který má nainstalován Internet Explorer v.7 a Mozillu Firefox v. 3.0.6 (tj. v současnosti nejvyužívanější prohlížeče). Důsledkem odlišné schopnosti převést zdrojový kód do finální podoby bylo to, že program správně běžící na jednom, vykazoval v druhém prohlížeči určité nesrovnalosti. Zdrojem těchto komplikací bylo z největší části HTML, CSS, méně pak již Javascript a PHP.

Nicméně i ve chvíli, kdy programový kód byl korektní pro oba prohlížeče, správnou funkčnost aplikace to ještě nezaručovalo. Samotný prohlížeč má množství vlastních nastavení, které mohou zamezit správnému chodu.

Příklady:

- v IE se mapa nezobrazí při vysoké úrovni vlastního zabezpečení
- IE v. 6 nezobrazuje oproti svému nástupci průhledné PNG obrázky
- MF vypočítává velikost stránky v pixelech i s šíří pravého posuvného
- u MF nelze u objektů vypnout tisk pozadí
- rozdílná správa cache paměti, která mnohdy zamezí zobrazení změněného objektu

Veškeré nalezené „chyby“ s jejich možným řešením jsou popsány v aplikaci pod ikonou „info o aplikaci“  pod částí nazvanou „Chyby aplikace a jejich řešení“.

Program byl vyvíjen se základním předpokladem: možnost využití aplikace z libovolného PC připojeného k internetu, jehož součástí je „libovolný“ prohlížeč. Z tohoto důvodu bylo nutno striktně dodržovat její funkčnost na nejpoužívanějších prohlížečích (Internet Explorer a Mozilla Firefox). Mnohdy programátorsky velmi nesnadná práce, nicméně pro splnění hlavní podmínky velmi důležitá. Neshody Internetu Exploreru (mnohdy i mezi jednotlivými verzemi) a Mozillou Firefox jsou zkušenějším v oboru dobře známy. Striktní dodržování standartů konsorcia W3C Mozillou Firefox na rozdíl od mnohdy benevolentního přístupu Microsoftu Explorer, s tímto vším se musí programátor vypořádat. Bohužel zjišťování, proč některé věci v IE fungují a v MF ne, stojí nemalé, nejen časové úsilí.

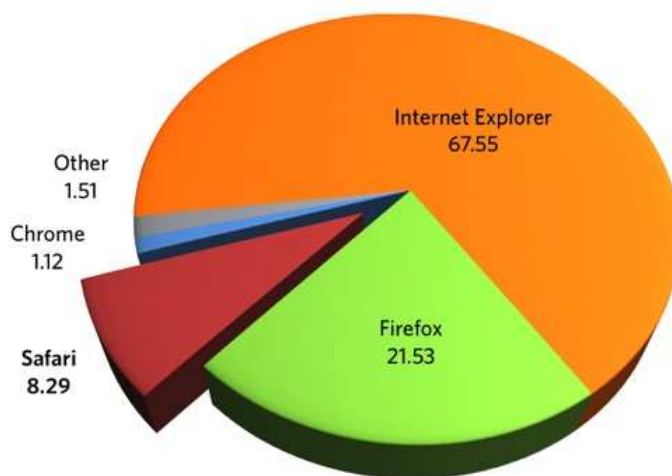
Na trhu statisticky stále bezkonkurenčně kraluje Internet Explorer. Je známo, že většina více počítačově vzdělaných uživatelů již IE nepoužívají a vyměnili ho za jiný. Nicméně filosofie společnosti Microsoft, kdy s instalací jejich operačního systému Windows se automaticky nainstaluje i prohlížeč IE, dělá své. Většina lidí (67,55 % - viz. Graf 4 Podíl prohlížečů v ČR) je rádo, že se jim spustí internet po kliku na kouzelné modré „éčko“. Že jsou schopni si vybrat email, přečíst zprávy, poslat SMS-ku. Tito lidé rozhodně nemají snahu zjišťovat, že na trhu existuje nějaký jiný, mnohdy sofistikovanější produkt a ten si „někde“ obstarat a nainstalovat.

Ve výsledku lze říci, že aplikace „Inventarizace starých ovocných sadů na území CHKO Bílé Karpaty“ funguje na všech nejpoužívanějších současných prohlížečích. IE v. 6, IE v. 7, MF, Opera. Nicméně práce je přizpůsobena prioritně prohlížečům IE v. 7 a MF.

Při testování aplikace byly zjišťovány i hodnoty využití operační paměti. Ve výsledku se nejednalo o žádné konkrétní hodnoty, nicméně s jistotou mohu tvrdit, že při paměťově náročné práci (např. načítáním polygonů) se po určitém čase IE „nafoukne“ až na 300 MB. Při celkové velikosti operační paměti počítače čítající 512 MB se jedná o kritickou hranici. Systém se lidově řečeno „seká“ a nezbývá nic jiného než mnohdy násilné ukončení prohlížeče. MF oproti tomu jakoby měla v tomto své limity. Není mi známa její technická specifikace, ale nikdy jsem se s ní i při zvyšující se náročnosti úkolů nedostal na více než 140MB. V tomto hodnocení se jasným favoritem stal prohlížeč Opera, která je založena více méně na obdobné platformě jako Mozilla Firefox (jednoduše lze říci, že většina věcí co funguje na MF, funguje i v Opeře). Hodnota využití operační paměti u Opery při práci se zastavila pod hodnotou 50 MB.

Net Applications browser market share: January '09

Percent



Source: Net Applications



Graf 4 Podíl prohlížečů v ČR [W21]

10.5. Výsledná aplikace - co obsahuje a jak pracuje

Hlavní cíl práce spočíval ve vytvoření aplikace, která bude zaznamenávat do mapy přesné polohy jak ovocných stromů, tak sadů. K tomuto jádru však bylo potřeba vytvořit přídatné součásti. Finální program je v podstatě kartografické dílo, které musí v rámci možností splňovat určité požadavky na mapu jako takovou kladené.

10.5.1. Generalizace

Databáze programu již nyní obsahuje 845 bodových (stromy) a 27 polygonových (sady) záznamů. Stromy jsou mnohdy v území velmi hustě koncentrované. Tento shluk je potřeba kartograficky vyřešit určitým stupněm **generalizace**. Vývojáři z Googlu na toto pamatovali funkcionalitou, která dokáže jednotlivé body zobrazovat (a skrývat) v závislosti na stupni přiblížení mapy (zoomu). Jak dále s touto možností naložit záleží jen na programátorovi.

Aplikace „Inventarizace ovocných stromů a sadů“ pracuje ve dvou („třech“) úrovních generalizace. Princip spočívá v následném: při zaškrtnutí tlačítka **okolí** (Obr. 17 Možnost zaktivovat bodovou generalizaci ... nezaškrtnutím jsou načteny a zobrazeny všechny body dle jejich přesné polohy) spolu s výběrem požadovaného druhu stromu se spočítá nejvhodnější úroveň

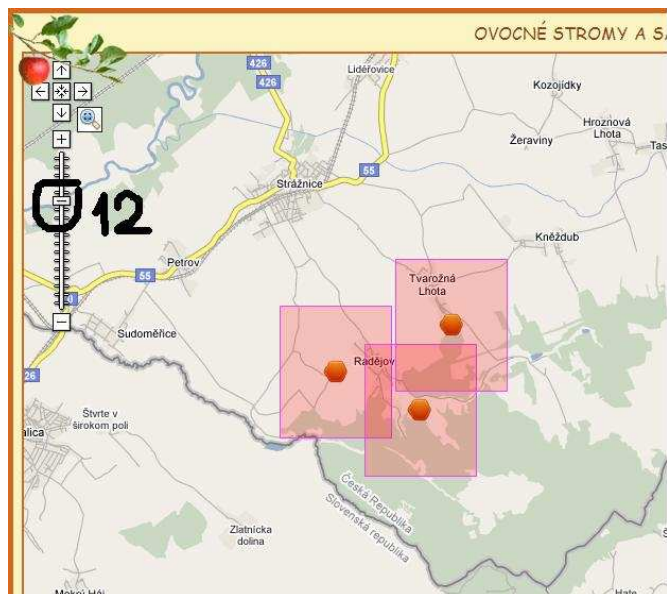
přiblížení (tematická mapa: 0-17 úrovní, satelitní snímek: 0-19 úrovní; pozn. 0-zobrazení celého světa, 17-detail) tak, aby mapové okno zároveň zobrazovalo všechny načtené body. Pro generalizaci jsem zvolil dvě úrovně. První je v rozmezí zoomu 0-13, druhá 14-15. Pokud je uživatel na úrovni v prvním či druhém rozmezí, stromy nebudou zobrazeny. Výskyt stromu bude v mapě zaznačen pouze symbolickou ikonou, kdy každá takováto má své plošné rozpětí, pod které spadají stromy okolní (viz. Obr. 18, 19, 20).



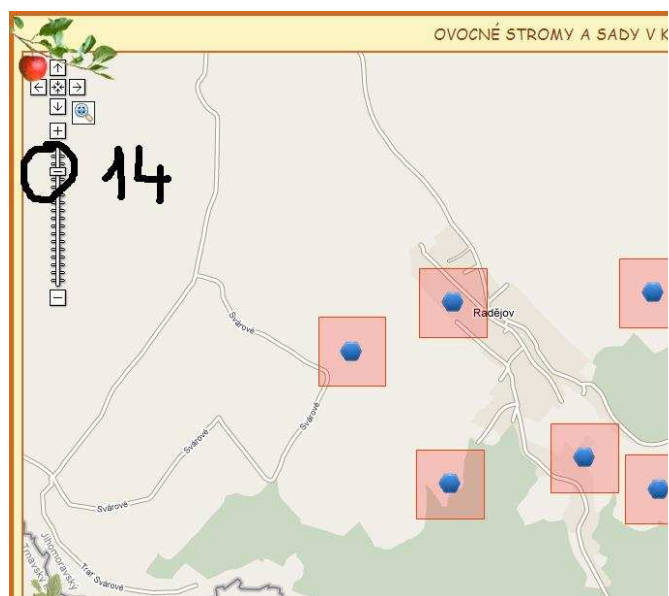
Obr. 17 Možnost zaktivovat bodovou generalizaci

Tj. pro první úroveň (0-13) se jedná o plochu zhruba 1,6 x 2,1km, druhá úroveň (14-15) 0,2km x 0,3km. Pokud se však již dostane uživatel nad zoom úroveň 15, symbolická ikona (ikony) se skryjí a zobrazí se stromy, které spadaly právě pod plošné rozpětí tohoto zástupného znaku. Konkrétní příklad:

*V terénu existují tři hrušky vzdáleny 5 m od sebe. Pakliže je chci zobrazit s určitým nastavením generalizace (zatržené políčko „**okolí**“), ikony jednotlivých hrušek se zobrazí pouze pokud budu na úrovni přiblížení 16 a více. Pokud se mapa oddálí pod úroveň 16, ikona hrušky zmizí a nahradí se zástupnou ikonou, která pohltí (skryje ikony sousedících hrušek) vše v okolí 0,2 x 0,3 km. Pokud uživatel mapu oddálí pod úroveň 14, změní se zástupný symbol za jiný, nicméně také zástupný. V tomto případě již s podstatně větším okolím, 1,6 x 2,1 km. Všechny prvně zástupné symboly spadající do aktuálního rozmezí zmizí.*



Obr. 18 Ukázka bodové generalizace – zoom úroveň 12



Obr. 19 Ukázka bodové generalizace – zoom úroveň 14




Obr. 20 Ukázka bodové generalizace – zoom úroveň 16

Generalizace složitého tvaru polygonu (sadu) probíhá dle Douglas-Peucker algoritmu (viz. 10.2.Data).

10.5.2. Legenda

Použité znaky a barevné rastry jsou popsány v **legendě**, která je spustitelná pomocí ikony pod mapovým polem .


10.5.3. Výpis záznamů

K jednotlivým stromům a sadům je mnohdy potřeba přistupovat jinak, než jen v grafickém prostředí. Bylo potřeba vyvinout přímé napojení do databáze a umožnit jejich textový **výpis** . Výpis s možností řazení dle uživatelem navolených podmínek. Následně je umožněn export těchto položek do textového souboru, kde jsou jednotlivé položky záznamu od sebe navzájem odděleny tabulátorem. S takto uspořádaným souborem se již dá dále pracovat. Krom standardních importů a následně např. tvorbě grafů, apod. se soubor dá využít i pro práci v jiném GISu. Pravdou je, že import do jiného geoinformatického programu není zas až tak jednoduchý, nicméně pro zkušenějšího uživatele by to neměl být zas až takový problém.


Návod k importu do ArcView 3.1: vyexportovaný textový soubor importovat v (např.) MS Excelu, zde „uložit jako“ DBF a následný import do ArcView. Zde již nic nebrání použití skriptu, který ze sloupců hodnot souřadnic vytvoří pole ID potřebné pro grafické zobrazení bodů v prostředí ArcView...ve své podstatě se jedná o popis principu vytvoření SHP souboru.

Pozn. uložení záznamů z aplikace „Inventarizace starých ovocných sadů ...“ do textového souboru na lokální disk je dostupné přes zdrojový kód načtené webové stránky (viz. MANUÁL - záznamy v textové podobě) – princip jistě trochu méně uživatelsky příjemný, nicméně vycházející ze základní myšlenky PHP. Daleko přívětivější řešení by spočívalo např. v kliku na potencionálně připravené tlačítko „ulož na disk“ a následné vybrání místa na disku. „Bohužel“ PHP programovací jazyk probíhá na serveru a tudíž není možné, už jen z bezpečnostního hlediska, aby tento jazyk dokázal manipulovat jakýmkoli způsobem se soubory na lokálním disku. Nebýt tohoto základního omezení, dal by se velmi lehce zneužít obsah uživatelského disku, třeba už jen po kliku na nějakou neznámou ikonu na internetové stránce.

10.5.4. Google Earth

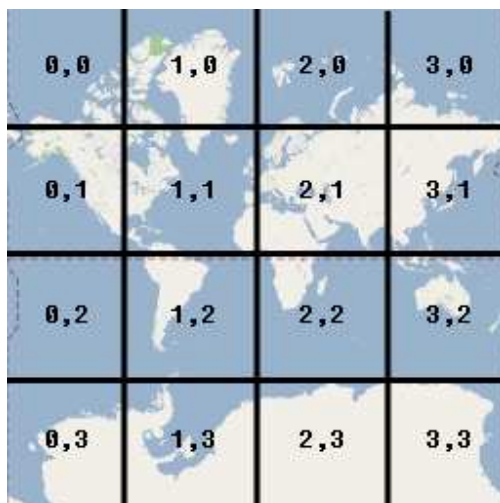
Google API nekončí svou mapovou funkcionalitu pouze u statického mapového portálu známou ze stránek maps.google.com. Jejím dalším velmi povedeným a na trhu prvním (zdarma poskytnutý s pokryvem celého světa) v oboru převratným dílem bylo vytvoření mapového portálu umožňujícího zobrazení Země v 3D perspektivě. Aplikace jménem **Google Earth** se uživatelům nabízí podobně, jako je tomu u statických Google Maps, přes rozhraní API. Funkcí pro ovládání této 3D mapy sice ještě zdaleka není tolik, kolik toho Google nabízí u statických map, nicméně už samotný krok otevření této brány je velký posun ve free GIS. Plnohodnotná možnost náhledu na 3D mapu Země je k dispozici po kliknutí na ikonu umístěnou vpravo dole pod mapovým oknem . Toto prostředí však s aplikací „Inventarizace ...“ nijak nesouvisí. Uživatelé však může pomáhat jako názorná ukázka terénu CHKO Bílých Karpat z jiné perspektivy, než jakou mu nabízí statické Google Maps.

10.5.5. Tisk

Podstatnou funkcí aplikace je **tisk** . Jako u každého „gis“ systému, tak i zde je potřeba výstupu do analogové podoby (pozn. k digitálnímu výstupu je nejvhodnější využít tlačítka pro sejmoutí obrazovky – Print Screen). Napsat jednoduchou proceduru přístupu k tiskárně nebylo v JavaScriptu tak složité. Určité komplikace nastali opět u vzájemné kompatibility prohlížečů. Ukázalo se, že např.

- IE, stejně tak i MF netisknou info údaje o prvku (bublinu)
- pozadí se u některých HTML prvků tiskne ve chvíli, kdy se v prohlížeči (konkrétně IE, u MF jsem obdobnou volbu nenašel) zaktivuje volba „tisk pozadí“
- při tisku v IE se nezobrazí červená šipka ukazující na vybraný sad – uživatel si vytisknuté údaje k objektu musí přiřadit sám
- MF při tisku zobrazuje bílé pruhy v mapě na místech, kde se skládají jednotlivé mapové dlaždice – místa v Obr. 21 naznačena černými liniemi (technická poznámka: mapa území je složena z obrázků


navzájem vedle sebe seskládaných; různá zoom úroveň=různě hustá síť čtverců -> princip rychlého načítání-AJAX)



Obr. 21 Popis a umístění jednotlivých dlaždic [W22]

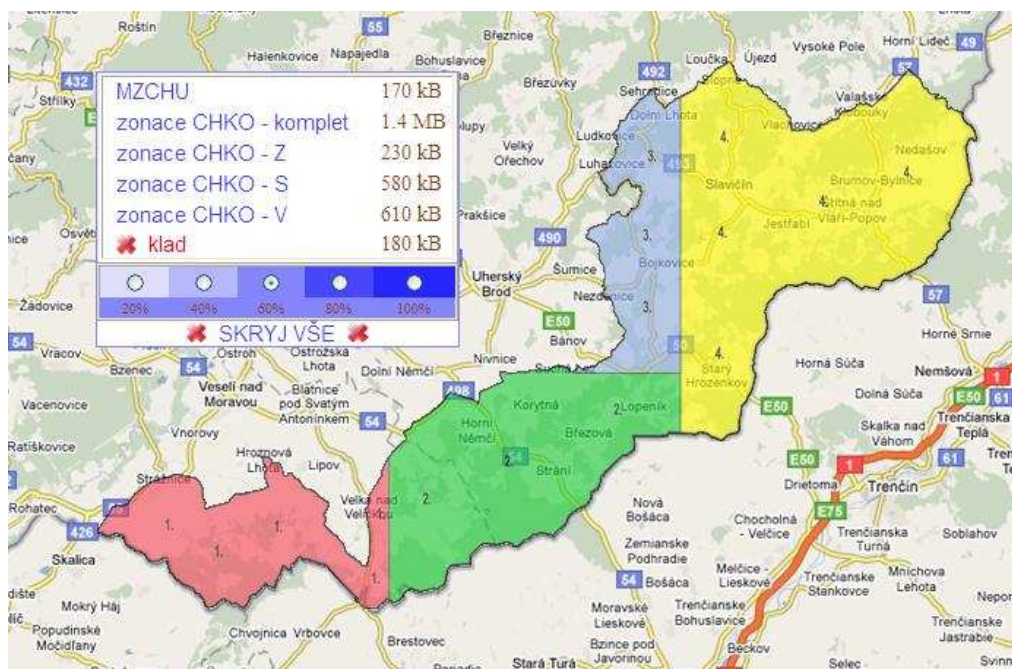
Závěrem pro shrnutí funkce TISK je potřeba podotknout, že tisk je „relativně bezproblémový“ z prohlížeče MF. V IE jsou drobné chyby, které však finální grafický výstup neznehodnotí.

10.5.6. Info o aplikaci

Pod ikonou  **info o aplikaci** je vytvořena webová stránka se stručným shrnutím jednotlivých možností programu s jejich popisem.

10.5.7. Přidání externí vrstvy

Ač Google Maps poskytují velmi kvalitní snímky ČR (letecké jsou srovnatelné v současnosti s nejkvalitnějšími volně dostupnými snímky na portálu www.mapy.cz; topografické v detailu měst zaostávají) je potřeba mnohdy připojit k stávajícímu podkladu **externí vrstvu** (Obr. 22). V GISu pojem známý jako WMS služba. Aplikace byla tvořena pracovníkům CHKO Bílých Karpat „na míru“, kde bylo dopředu známo, jaké vrstvy budou potřeba. Díky tomu nebylo potřeba vyvíjet prostředí pro WMS službu, ale stačila možnost vložení jednotlivých map přímo do mapového podkladu, aniž by se musela tato služba inicializovat. Tyto vrstvy („klad“, „zonace CHKO“, „Maloplošná zvláště chráněná území“) jsou možno přidat po kliku pravým tlačítkem myši do mapy. Vrstvy je možno vložit s nastavitelným procentem průhlednosti.




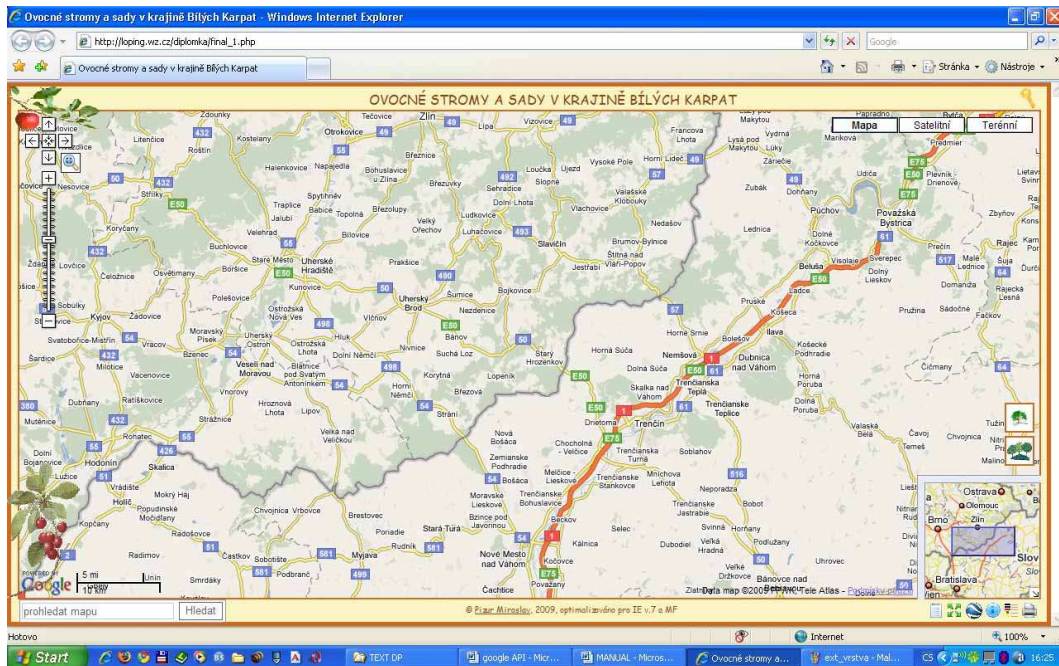
Obr. 22 Přidání externí vrstvy

Pozn. po domluvě s budoucími uživateli aplikace jsou posledně dvě jmenované vrstvy možné připojit až v administrátorském prostředí („zonace CHKO“ se často lidem, kteří netušili, co tato vrstva znamená, máta s vrstvou „klad“).

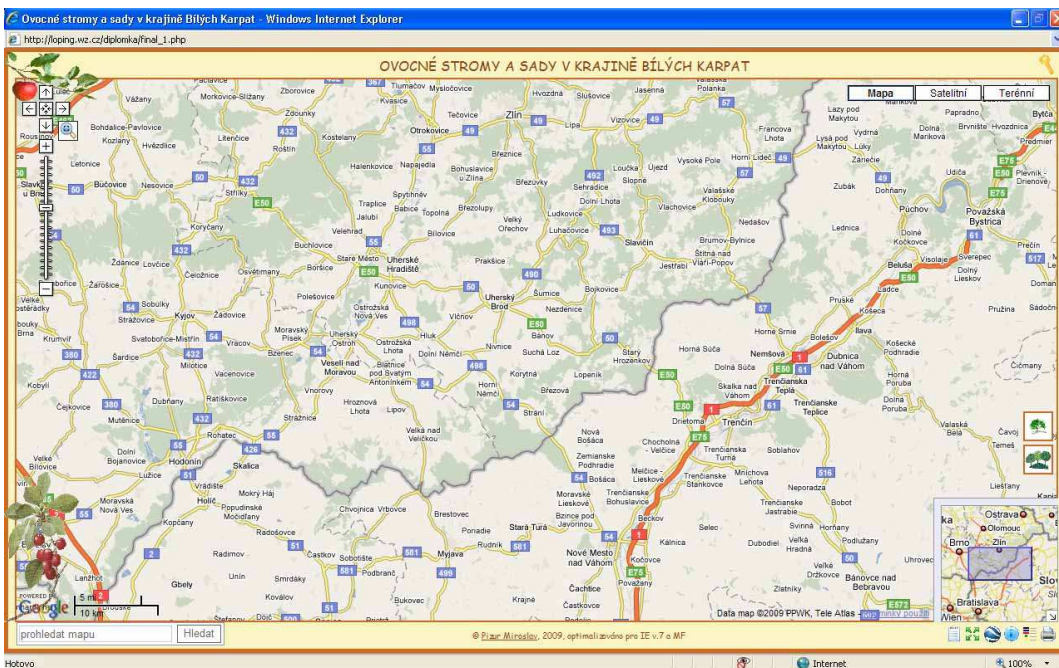
10.5.8. Přes celé okno

K další uživatelsky zpříjemňující vlastnosti aplikace patří ikona **přes celé**

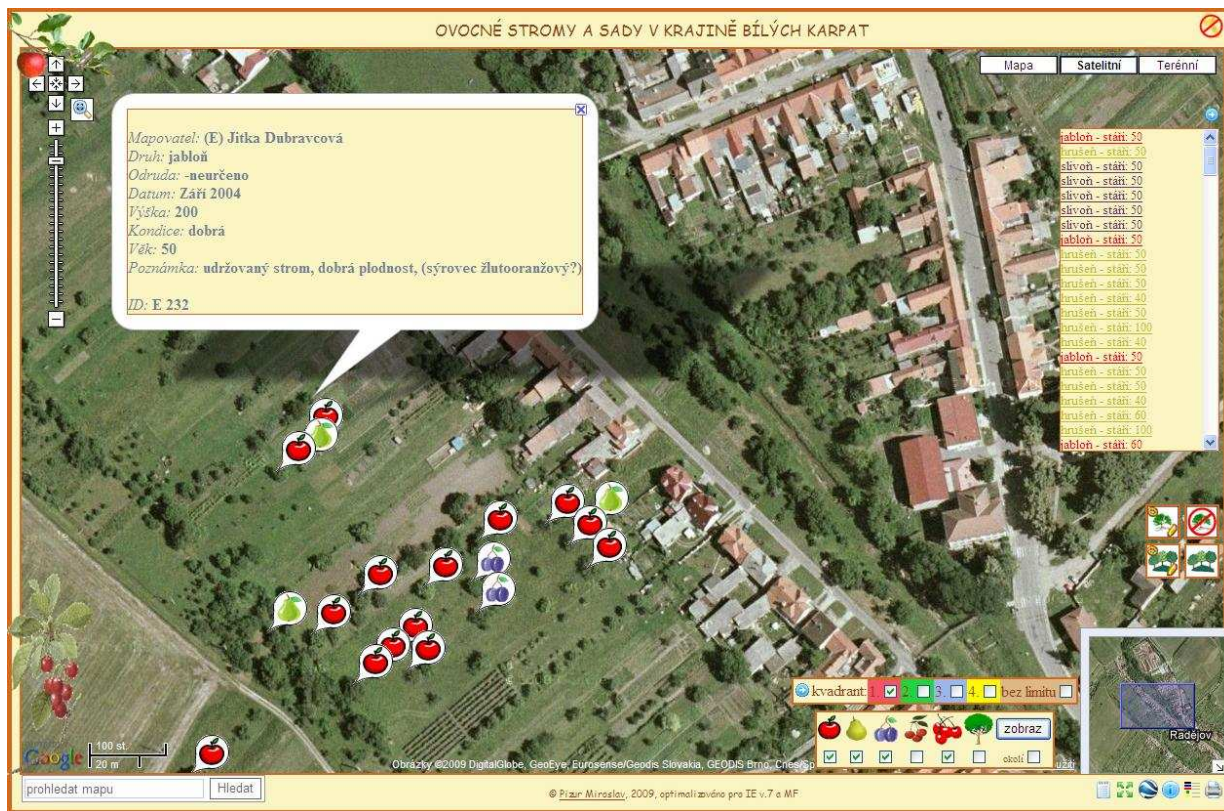
okno . Po její aktivaci se viditelné prostředí aplikace zobrazí do okna nového s tím, že toto okna má již maximální možnou velikost. Každý prohlížeč na tento příkaz opět reaguje trochu jinak. Společným rysem však zůstává snaha o co možná největší maximalizaci mapového okna. Tzn. schování všech „nedůležitých“ ikon, menu položek, apod.



Obr. 23 Standardní zobrazení aplikace v IE



Obr. 24 Maximalizace zobrazitelné plochy v IE



Obr. 25 Ukázka aplikace

Internetová adresa aplikace:

sady.webz.cz




10.6. Editační prostředí


Druhou velkou částí funkcionality aplikace je její editační prostředí (popsáno v MANUÁL – Administrátorská funkcionality) dostupné po zadání přihlašovacího jména a hesla. Ověření hesla pracuje na principu uložení zaheslovaného řetězce do databáze metodou MD5. Heslo zadané uživatelem se po odeslání zašifruje stejnou metodou, kde se porovná s originálem v databázi. V případě shody se aplikace přepne do administrátorského režimu, v případě opačném se uživatel vyzve k návratu na původní stránku určenou pro hosty.

10.7. Ovládání aplikace

Jednou z nejdůležitějších priorit při vývoji aplikace byl důraz na uživatelskou příjemnost. Práce je určena pro mapovatele, ne pro specialisty pracující s GIS systémy. Tito lidé většinou nemají s GISem velké zkušenosti. Práce musí být intuitivní, přehledná, uživatelsky přívětivá a přitom z technického hlediska funkční.

Téměř veškerá práce v aplikaci se nějakým způsobem vztahuje k mapě. Z toho pramení logická snaha o zobrazení mapy do maximální možné velikosti. Technické řešení tohoto principu spočívá v změření zobrazitelné plochy na monitoru a následném udání relativní procentuální velikosti.

- ovládání aplikace je kompletně řešeno grafickými názornými ikonami, ke kterým byly přidány popisky
- pro nastavení datumu bylo využito externí javascriptové aplikace, kde si uživatel jednoduše kliknutím vybere požadovaný datum
- přidávání vrstev pracuje na systému plovoucího okna, ke kterému se uživatel dostane přes pravý klik do mapy
- pro přibližovací (zoomovací) fce byly přidány krom standardních google prvků navíc výběrový obdélník  (možnost vybrat a přiblížit přesně požadovanou oblast v mapě) a rolovací kolečko myši
- přesná lokalizace hledaného místa je řešena skrze Google vyhledávač - do textového pole lze zadat požadovaný název hledaného místa, kde po vybrání nalezeného výsledku se mapa sama nastaví nad hledaný cíl
- dialogové okno je možno schovat kliknutím na tlačítko  a zpětně zobrazit tlačítkem 

Ač je systém koncipován s prvořadou snahou o srozumitelnost, přeci jenom obsahuje určité množství funkcí, jejichž ovládání jsou mnohdy pro uživatele na první pohled nejasné. Z tohoto důvodu byl vytvořen k aplikaci pomocník  **info o aplikaci**, kde je jednoduše popsáno, jak se k požadovanému cíli pracovat.

Plný návod popisující aplikaci je popsán v **MANUÁL** a slouží jako samostatná příloha této diplomové práce.

10.8. Diskuse o nekompaktnosti prohlížečů - co musí být nastaveno

Prohlížeče mají v současnosti opravdu velmi širokou paletu různých bezpečnostních nastavení. V dobách elektronického bankovníctví, virů, spamů a vyskakujících oken se není ani čemu divit. Aplikace „Inventarizace ovocných sadů ...“ využívá některé prvky, které si vyžadují určitou bezpečnostní benevolenci.

V této kapitole asi nepokryji všechny možné úskalí, kdy a za jakých bezpečnostních nastavení jak PC, tak samotných prohlížečů aplikace pracuje. Bude zde uvedeno alespoň z mnou provedeného průzkumu základní rámec problémů a doporučených nastavení.

Pozn. průzkum probíhal na 15-ti na sobě nezávislých počítačích, kde každý využíval různorodý prohlížeč s variabilně nastaveným zabezpečením PC - spyware, antivir; prohlížeče byly vždy IE či MF různých verzí.

Zjištěných problémových nastavení není velké množství. Problémy nastávaly v podstatě pouze při nestandardních a většinou uživatelem nějak pozměněných úrovních zabezpečení.

- ve chvíli, kdy prohlížeč (testováno na IE) má nastavenou VYSOKOU ÚROVEŇ ZABEZPEČENÍ, podkladové mapy Googlu se nezobrazí; je potřeba úroveň snížit minimálně na střední – standardní úroveň.
- aplikace neběžela na PC v IE kde byl nainstalován program na ochranu proti spywaru, konkrétně „Spybot S&D“ – ve chvíli kdy byla ve zmiňovaném programu vypnuta volba speciální ochrany IE, aplikace (konkrétně načítání stromů) pracovala správně

11. ZÁVĚR

Technologie připojení prostřednictvím API k mapám různých společností je v současnosti velmi dynamicky se rozvíjející oblastí. Tyto technologie staví na v praxi ověřených, známých a hojně využívaných mapových systémech. Jejich relativně jednoduchá dostupnost posouvá současný GIS zase o krok dále. Již dnes existuje množství free GIS systémů, nicméně jen málokterý z nich se dostane do povědomí široké laické veřejnosti. Problém nespočívá v tom, že by snad tyto systémy nebyly kvalitní, problém spočívá v osvětě a schopnosti ukázat veřejnosti, že užitek z něj mohou mít i oni. Odborníkům toto vysvětlovat není potřeba, ti si většinou sami zjistí, jaký software je pro jejich úkol vhodný a který zase ne. Ale veřejnost? A ta se oproti fundovaným odborníkům řídí úplně jinými hledisky. Veřejnost upřednostňuje jednoduchost a grafickou stránku nad funkčností... A zde bych viděl snad největší pozitivum této nové technologie. Programátor má nyní k dispozici plně funkční, přívětivý a velmi známý nástroj na jehož základě tvoří. Jak se mu výsledné dílo povede, záleží již jen a jen na něm samotném. A že slova o „přívětivosti, oblíbenosti...“ nejsou jen reklamními slogany, tak často používané, svědčí realita. V současnosti jsem v zaměstnaneckém poměru, kde mým úkolem je kontakt s širokou laickou veřejností nějak s počítačem svázanou (úředníci, poštovní úřady, účetní,...). Ve chvíli, kdy jsem jim položil otázku, zda již někdy pracovali s internetovými mapami, skoro všichni se na mě podívali s mírně ironickým úsměvem a dovětkem „tak hloupí ještě nejsme“.

Google Maps API jsou v současnosti v tomto mladém odvětví svou vyspělostí na prvním místě. Jeho tvůrci velmi hbitě pracují na jeho slabých místech a vytváří nespočet nových rozšíření. Nicméně co se společnosti Google v současnosti daří asi ze všeho nejvíce, je spojení kapitálu, technologií a síly na IT trhu. Díky těmto třem faktorům si tvůrci mohou dovolit vypustit do světa většinu zdrojových kódů svých produktů. Zde na ně již čekají tisíce programátorů, kteří participují na těchto otevřených základech. Ve výsledku vznikají nové aplikace, které jsou založeny právě na platformě Google. Touto filosofií si firma upevňuje jednu z předních příček nejen na trhu GIS produktů.

S Google Maps API se dají vytvořit opravdu velmi rychle jednoduché mapové aplikace. Nicméně pořád je to programování. Tzn. úroveň hotového programu je úměrná úrovni znalosti dotyčného tvůrce. Společnost Google dává k dispozici opravdu silný a mohutný nástroj, avšak plně ovládnout ho dokáže opravdu jen hrstka zkušených IT odborníků.

Ve chvíli, kdy je potřeba vytvořit nějakou složitější aplikaci, závislou na mnoha dílčích detailech (např. kartografické zásady), složitost zdrojového kódu úměrně roste. Není problém z ukázkových příkladů vykopírovat

algoritmus pro načtení bodu. Nicméně je již daleko složitější tento bod ošetřit tak, aby se vložil do mapového okna pouze ve chvíli, kdy je uživatel přihlášený jako administrátor, aby se neukládaly změny polohy bodu, pakliže není kliknuto na tlačítko „ulož“, aby se mapový podklad přizpůsobil určitým, uživatelem vyfiltrovaným souborem bodů. Aby se polygon ukládal v zašifrované podobě a přitom byl kdykoli schopen se rozšifrovat a připraven k editaci. Aby český jazyk standardně nastavený českou kódovou stránkou spolupracoval s mapami americké společnosti... Vytvořit ve všech směrech funkčně provázanou aplikaci již opravdu není úkol úměrný základní úrovni počítačového uživatele.

Krom nesporného faktu, že vývoj ucelenější a propracovanější aplikace vyžaduje určitý stupeň programátorských schopností, spočívá určité „omezení“ také v technologii komunikace. Většina systémů mapových API je založena na programovacím jazyku Javascript. Bohužel je to právě tento jazyk, který není i pro zkušené programátory pro svou „jednoduchost“, nicméně častou nevyzpytatelnost zrovna oblíbený...Nyní již mohu hodnotit i z vlastní zkušenosti. Víím, že stavět takovouhle „složitou“ práci na tomto programovacím jazyku není nic jednoduchého.

Na portfoliu znalostí potřebných pro dokončení této aplikace by se dala pěkně demonstrovat potřeba na současné GIS specialisty. Pakliže budu hodnotit čistě mnou vynaložený čas strávený nad vypracováním, pak mohu s jistotou tvrdit následné. 95 % veškerého stráveného času při vývoji mi zabralo samotné programování a práce s databázemi. Zbývajících 5 % zůstalo na kartografické záležitosti, kdy jsem si ku příkladu promýšlel, jak kartograficky správně zgeneralizovat větší množství stromů.

Ač jsem vytvořením této práce „obětoval“ mnoho hodin, nelituji. Dovoluji si s čistým svědomím a vědomím tvrdit, že kdybych neprodloužil dokončení studia o rok, tak bych tuto práci nikdy neodevzdal (minimálně) v takovéhle podobě v jaké v současnosti je. Nicméně nelituji! Byl jsem „donucen“ se ponořit do úplně nového světa, do světa praktického, toho, kde se GIS vyvíjí. Nechci své nabyté znalosti přeceňovat, nicméně ač jsem do této sféry nakoukl opravdu jenom úzkou škvírou, můj pohled se na mnoho GIS programů a jejich technické provedení změnil. Dnes již pracuji na městském úřadě a mým úkolem je zavádění eGovernmentu do státní sféry. Programy kolem spisových agend, datových schránek a také částečně mapových portálů jsou toho základním stavebním kamenem .. nicméně já již nyní na konferenci rozumím informatikům vysvětlující jádro nově připravovaných a nabízených systémů. Díky nynějším znalostem se do problematiky z informatického hlediska dostávám daleko rychleji, než mí kolegové „úředníci“.

12. LITERATURA

Knihy a jiné zdroje

- [1] Gibbon, R., Schuyler, E. (2006): Google Maps Hacks. O' Reilly, Sebastopol, CA, 321 s.
- [2] Lewis, A., Purvis, M., Sambells, J., Turner, C.(2007): Beginning Google Maps Applications with Rails and Ajax. Springer-Verlag New York, 357 s.
- [3] Williams, H. E., Lane, D. (2002): PHP a MySQL. Computer Press,Praha, 552 s.
- [4] Flanagan, D., (2002): JavaScript: kompletní průvodce. Computer Press, Praha, 825 s.
- [5] Hlavenka, J., Sedlář, R., Holčík, T., Šebesta, M., Botík, R., (1997): Vytváříme www stránky. Computer Press Praha, 393 s.
- [6] Ondra, R.: Testování API mapových služeb a tvorba online GIS aplikace. Diplomová práce. VŠB Ostrava, 2008
- [7] Sochorová, M.: Studie a zhodnocení zahraničních volně dostupných API mapových služeb. Diplomová práce. VŠB Ostrava, 2008
- [8] Holzner, S.: JavaScript profesionálně. Praha : Mobil Media a.s., 2003. 1071 s.
- [9] ZBIEJCZUK, Adam. Web 2.0-charakteristika a služby. [s.l.], 2007. 71 s. Diplomová práce.
- [10] DRUSKA, Petr. CSS a XHTML : Tvorba dokonalých webových stránek krok za krokem. [s.l.] : GRADA Publishing, a.s. , 2006. 200 s. ISBN 8024713829.
- [11] LANGRIDGE, Stuart. DHTML Utopia : Modern Web Design, Using JavaScript & DOM. Collingwood Australia : Sitepoint Publishing, 2005. 317 s. ISBN 0957921896.
- [12] HOLZNER, Steven. JavaScript profesionálně. Praha : Mobil Media a.s., 2003. 1071 s. ISBN 8086593401.
- [13] ŠKULTÉTY, Rastislav. JavaScript : Programujeme internetové aplikace. 2. aktualiz. vyd. [s.l.] : Computer Press, 2004. 224 s.

- [14] HOLZENER, Steven. Mistrovství v AJAXu : Naučte se programovat moderní aplikace. Jakub Zemánek. Brno : Computer Press, 2007. 590 s. ISBN 9788025118504.

Internetové zdroje

- [W1] Jak psát web – o tvorbě internetových stránek [online]. c1996, poslední revize březen 2009 [cit. 2009-1-10]. <<http://www.jakpsatweb.cz>>.
- [W2] Google maps API [online]. c2005, poslední revize duben 2009 [cit. 2009-2-21]. <<http://code.google.com/intl/cs/apis/maps>>.
- [W3] Google Maps API Tutorial [online]. c2009, [cit. 2009-1-30]. <<http://econym.googlepages.com>>.
- [W4] Encoding polylines for Google Maps [online]. c2006, poslední revize srpen 2006 [cit. 2008-6-18]. <<http://facstaff.unca.edu/mcmclur/GoogleMaps/EncodePolyline>>.
- [W5] Javascript tutorial [online]. c1999, poslední revize únor 2009 [cit. 2009-1-15]. <<http://www.w3schools.com>>.
- [W6] PHP tutorial [online]. c2001, poslední revize duben 2009 [cit. 2009-2-13]. <<http://php.net>>.
- [W7] Programmableweb : Mashup [online]. c2008 [cit. 2008-03-20]. Dostupný z WWW: <<http://programmableweb.com/mashups>>.
- [W8] JAMES, Garrett. AJAX : A New Approach to Web Applications [online]. c2008 [cit.2008-03-25]. <<http://adaptivepath.com/ideas/essays/archives/000385.php>>.
- [W9] Google Suggest [online]. c2006, [cit. 2008-03-25]. <<http://www.google.com/webhp?complete=1&hl=en>>.
- [W10] Mapový portál - Seznam [online]. c2009 [cit. 2009-3-15]. <<http://www.mapy.cz>>.
- [W11] Mapový portál - Atlas [online]. c2009 [cit. 2009-3-15]. <<http://mapy.atlas.cz>>.
- [W12] Getting Started with the Map Control [online]. c2008. <<http://msdn2.microsoft.com/en-us/library/bb429637.aspx>>.

- [W13] Mapy Google [online]. c2008. <<http://maps.google.com/>>.
- [W14] Mapy Yahoo! [online]. c2008. <<http://maps.yahoo.com/>>.
- [W15] Google Maps API – diskuzní skupiny [online]. C2009, poslední revize duben 2009 [cit. 2009-12-30].
<<http://groups.google.com/group/Google-Maps-API/topics>>.
- [W16] Gug.cz - Vývojáři [online]. c2009, poslední revize březen 2009 [cit. 2009-3-13]. <<http://groups.google.com/group/gugcz-vyvojari>>.
- [W17] Bílé Karpaty [online]. c2009, poslední revize březen 2009 [cit. 2009-1-10]. <<http://www.bilekarpaty.cz/>>.
- [W18] Encoded Polyline Algorithm Format [online]. c2008, poslední revize červen 2006 [cit. 2008-5-13].
<<http://code.google.com/intl/cs/apis/maps/documentation/polylinealgorithm.html>>.
- [W19] The encoding algorithm for the levels string [online]. c2008, poslední revize srpen 2006 [cit. 2008-5-13].
<<http://facstaff.unca.edu/mcmclur/GoogleMaps/EncodePolyline/algorithm.html>>.
- [W20] Interactive Polyline Encoder Utility [online]. c2008, poslední revize červen 2006 [cit. 2008-5-13].
<<http://code.google.com/intl/cs/apis/maps/documentation/polylineutility.html>>.
- [W21] Výsledky průzkumu – podíl prohlížečů [online]. c2009, poslední revize únor 2009 [cit. 2009-2-9].
<<http://www.mujsmac.cz/art/hw/netapplications-leden09-4-2-09.html>>.
- [W22] Map Overlays – Custom Map Types [online]. c2009, poslední revize duben 2009 [cit. 2009-2-4].
<http://code.google.com/intl/cs/apis/maps/documentation/overlays.html#Tile_Overlays>.
- [W23] London executive [online]. c2008,
<<http://www.londonexecutive.com/propertybird/>>.
- [W24] Dual Maps [online]. c2007,
<<http://www.mapchannels.com/DualMaps.aspx>>.

- [W25] Zaznamenání života v San Francisku pomocí videí [online]. c2008, <<http://www.angieandderek.com/videos/>>.
- [W26] Dual Maps [online]. c2007, <<http://www.mapchannels.com/DualMaps.aspx>>
- [W27] HOROVČÁK, Pavel. Webové služby - třetí generace internetu. IT Systems [online]. 2003, roč. 9 [cit. 2008-03-31].
- [W28] Open Geospatial Consortium [online]. c1994-2008 [cit. 2008-04-01]. Dostupný z WWW: <<http://www.opengeospatial.org/>>.
- [W29] XHTML2 Working Group Home Page [online]. c1995-2007 [cit. 2008-02-25]. Dostupný z WWW: <<http://www.w3.org/Markup/#tutorials>>.
- [W30] CSS Tutorial [online]. 2003 [cit. 2008-02-25]. Dostupný z WWW: <<http://www.html.net/tutorials/css/>>.
- [W31] JANOVSKEÝ, Dušan. CSS styly [online]. [2004] [cit. 2008-02-25]. Dostupný z WWW: <<http://www.jakpsatweb.cz/css/>>.
- [W41] Document Object Model Core [online]. 2000 [cit. 2008-03-01]. Dostupný z WWW: <<http://www.w3.org/TR/DOM-Level-2-Core/core.html>>.
- [W32] ABRATT, Deborah, MALLINSON, Wayne, BEKKER, Antonet. Usability Testing: Recipe for Success [online]. 2003 [cit. 2008-03-10]. Dostupný z WWW: <http://www.testingstandards.co.uk/usability_recipe_for_success.htm>.
- [W33] Google Map API Concepts [online]. Dostupný na WWW: <<http://code.google.com/apis/maps/documentation/index.html>>
- [W34] Google Maps API Reference [online]. Dostupný na WWW: <<http://code.google.com/intl/cs-CZ/apis/maps/documentation/reference.html>>

13. SUMMARY

Title of the thesis: Website for old orchard inventory in Bile Karpaty area

Website for old orchard inventory in Bile Karpaty area - have for one's object to create a web portal that will allow the old orchards inventory. The portal must offer tools for creating a new record, including its own location, management and presentation of contents data. The portal must be able to deal with stereometric part, as well as the descriptive part of stocktaking record. The final product should be built on minimal possible hardware, software and supporting data requirements.

SEZNAM PŘÍLOH

- MANUÁL
- CD-ROM
 - vstupní data (shp)
 - vstupní data po konverzi (sql)
 - zdrojové kódy vytvořené aplikace
 - www adresa aplikace
 - přístupové údaje do administrátorského rozhraní
 - podrobný popis dat na CD je v souboru readme.txt umístěném v hlavním adresáři tohoto CD