Univerzita Pardubice
Fakulta ekonomicko-správní

AJAX a jeho využití v GIS aplikacích

Přemysl Lédl

Bakalářská práce
2010

Univerzita Pardubice
Fakulta ekonomicko-správní
Ústav systémového inženýrství a informatiky
Akademický rok: **2009/2010**

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení: **Přemysl LÉDL**

Studijní program: **B6209 Systémové inženýrství a informatika**

Studijní obor: **Informatika ve veřejné správě**

Název tématu: **AJAX a jeho využití v GIS aplikacích**

Zásady pro vypracování:

- Technologie AJAX a její využití v moderních webových aplikacích
- Posouzení vhodnosti AJAXu pro běžné funkce webových aplikací, zhodnocení výpočetní náročnosti na straně serveru i klienta
- Vytvoření vzorové webové GIS aplikace s vhodným využitím AJAXu

Rozsah grafických prací:

Rozsah pracovní zprávy:

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam odborné literatury:

KUČERA, Miroslav. HTML tipy a triky od profesionálů. Praha : UNIS Publishing, 2001. 78 s. ISBN 8086097641.

LACKO, Luboslav. Ajax Hotová řešení. Brno : Computer Press, 2008. 272 s. ISBN 9788025121085.

BRÁZA, Jiří. PHP 4 učebnice základů jazyka. Praha : Grada Publishing, 2002. 224 s. ISBN 8024704420.

Vedoucí bakalářské práce: **Ing. Martin Novák**
Ústav systémového inženýrství a informatiky

Datum zadání bakalářské práce: **5. října 2009**

Termín odevzdání bakalářské práce: **30. dubna 2010**

L.S.

doc. Ing. Renáta Myšková, Ph.D.
děkanka

doc. Ing. Jiří Křupka, Ph.D.
vedoucí ústavu

V Pardubicích dne 5. října 2009

Prohlašuji:

Tuto práci jsem vypracoval samostatně. Veškeré literární prameny a informace, které jsem v práci využil, jsou uvedeny v seznamu použité literatury.

Byl jsem seznámen s tím, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorský zákon, zejména se skutečností, že Univerzita Pardubice má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle § 60 odst. 1 autorského zákona, a s tím, že pokud dojde k užití této práce mnou nebo bude poskytnuta licence o užití jinému subjektu, je Univerzita Pardubice oprávněna ode mne požadovat přiměřený příspěvek na úhradu nákladů, které na vytvoření díla vynaložila, a to podle okolností až do jejich skutečné výše.

Souhlasím s prezenčním zpřístupněním své práce v Univerzitní knihovně.

V Pardubicích dne 20. 4. 2010

Přemysl Lédl

Poděkování

Tímto bych chtěl poděkovat Ing. Martinu Novákovi, vedoucímu mé bakalářské práce, za jeho pomoc při tvorbě této práce a čas, který mi věnoval v rámci konzultací.

ANOTACE

Tato bakalářská práce se zabývá technologií AJAX a souvisejícími technologiemi jako jsou technologie JavaScript a DOM. Dále popisuje historický vývoj technologie AJAX v kontextu celkového obecného vývoje internetových technologií. Práce se zaměřuje na základní popis technologie AJAX a její implementaci v internetových aplikacích. V této práci jsou dále teoreticky popsány výhody a nevýhody této technologie a nejvhodnější oblasti implementace. V poslední části práce je popsán návrh webového rozhraní GIS aplikace, která je postavena na technologii AJAX.

KLÍČOVÁ SLOVA

AJAX, JavaScript, DOM, XMLHttpRequest, GIS, MapServer

TITLE

AJAX and its usage in GIS applications

ANOTATION

This bachelor thesis deals with the AJAX technology and related technologies such as technologies JavaScript and DOM. Furthermore it describes the historical development of AJAX technology in the context of general development of Internet technologies. In addition, this thesis focuses on the basic description of AJAX technology and its implementation in the Internet applications. This is followed by a theoretical description of advantages and disadvantages of this technology and the most suitable areas of implementation which is based on AJAX technology.

KEYWORDS

AJAX, JavaScript, DOM, XMLHttpRequest, GIS, MapServer

Obsah

Úvod

V oblasti tvorby internetových a intranetových aplikací v současnosti dominuje požadavek, aby se chování internetových a intranetových aplikací do jisté míry přiblížilo chování aplikací klasických takzvaně desktopových. Možností k zajištění takovýchto vlastností je více. Technologie, které to umožňují, se objevily již dříve. Snaha vyvolat dojem desktopové aplikace v prostředí internetu není nikterak nová. Bohužel žádná z technologií nesplňovala potřebnou nezávislost a samostatnost. Vždy bylo zapotřebí nějakých dalších prostředků, které umožňovaly s takovou technologií pracovat. Důsledkem toho se objevila technologie nová, která se nazývá AJAX.

Jedná se o spojení dostupných a využívaných technologií, jež jsou obsaženy v samotném názvu AJAX. AJAX (Asynchronous JavaScript and XML) využívá asynchronní komunikaci mezi klientem a serverem, dále využívá značkovací jazyk XML (eXtensible Markup Language) a poslední a nejdůležitější technologií je skriptovací jazyk JavaScript. V tomto jazyce se skripty aplikace využívající technologii AJAX píší a umožňují tak využívat všechny vlastnosti skriptovacího jazyka JavaScript.

Tato bakalářská práce je koncipována pro dosažení tří hlavních cílů. Na základě definovaných cílů je práce strukturována do příslušných kapitol zabývajících se podrobně danou konkrétní oblastí, jež se vztahuje k technologii AJAX a její využití při tvorbě internetových a intranetových aplikací.

Prvním cílem práce je popsat technologii AJAX, vyzdvihnout její přednosti vzhledem k vývoji trendů souvisejících s tvorbou internetových aplikací a současně představit jednotlivé technologie a jejich úlohy. Tímto cílem se práce zabývá v první kapitole, která mimo jiné dále shrnuje historii vzniku technologie AJAX.

Druhým neméně důležitým záměrem práce je objasnit výhody a nevýhody použití technologie AJAX při tvorbě internetových aplikací. Na základě těchto skutečností popsat nejpoužívanější oblasti pro uplatnění této technologie i s přihlédnutím na zmiňované výhody a nevýhody. Dále také zhodnotit výkonnost aplikace při nasazení technologie AJAX oproti klasické aplikaci a posoudit vhodnost a efektivnost zvolené technologie AJAX v celkovém kontextu.

Posledním cílem této bakalářské práce, jemuž je vyčleněna poslední třetí kapitola, je navrhnout a popsat webové rozhraní GIS aplikace, která využívá technologie AJAX a zároveň zhodnotit výkonnost a přívětivost webové rozhraní GIS aplikace z pohledu uživatele.

Celkovým záměrem této bakalářské práce je:

- Představit technologii AJAX v kontextu rozvoje internetu a požadavků uživatelů internetu.
- Současně představit technologie, které tvoří technologii AJAX.
- Zdůraznit oblasti uplatnění a nasazení této technologie.
- Posoudit její výkonnost ve srovnání s klasickými přístupy.
- Navrhnout webové rozhraní GIS aplikace, která využívá technologie AJAX.
- Zhodnotit výběr technologie AJAX pro tvorbu webové rozhraní GIS aplikace z pohledu uživatele.

1 AJAX

V této kapitole je popsána historie vzniku technologie AJAX a současný trend internetových aplikací. Dále jsou popsány technologie využívané AJAXem.

1.1 Historie vzniku

S rozvojem internetových a intranetových aplikací se stále více zvyšují požadavky uživatelů na tyto aplikace. Veškeré změny a nové možnosti v této oblasti vyvolávají vyšší a náročnější požadavky uživatelů. V současné době je kladen největší důraz na to, aby se internetové aplikace co nejvíce podobaly klasickým desktopovým aplikacím. Nejrozšířenější technologie, které se využívají pro samotnou konstrukci, jedná se především o serverové skriptovací jazyky PHP (Hypertext Preprocessor), ASP (Active Server Pages), ASP.NET, mají zásadní nevýhody při realizaci těchto požadavků. Hlavní nevýhodou je nutnost načítat obsah aplikace při každé provedené změně. Dále také problém s udržením informací při využívání skriptovacích jazyků na straně klienta především JavaScriptu.

Určité uspokojení tohoto požadavku na internetové aplikace nabízí Applet, který je vytvořen v programovacím jazyce JAVA[1]. Jedná se o desktopovou aplikaci, která je implementovaná do internetové aplikace pomocí speciálních HTML tagů. Další možné řešení je technologie FLASH[2]. Ovšem tyto dvě technologie vyžadují určité další prostředky pro svoji funkčnost na straně uživatele. V případě Java Appletu je to Java Virtual Machine a pro využití technologie FLASH je zapotřebí Adobe flash player.

Jedním s možných řešení, které vyhovuje novodobým požadavkům uživatelů internetových aplikací, je technologie AJAX. Jedná se o relativně mladou technologii. „Poprvé se tento pojem objevil začátkem roku 2005 ve článku Jesse James Garretta: A New Approach to Web Applications[9]“.

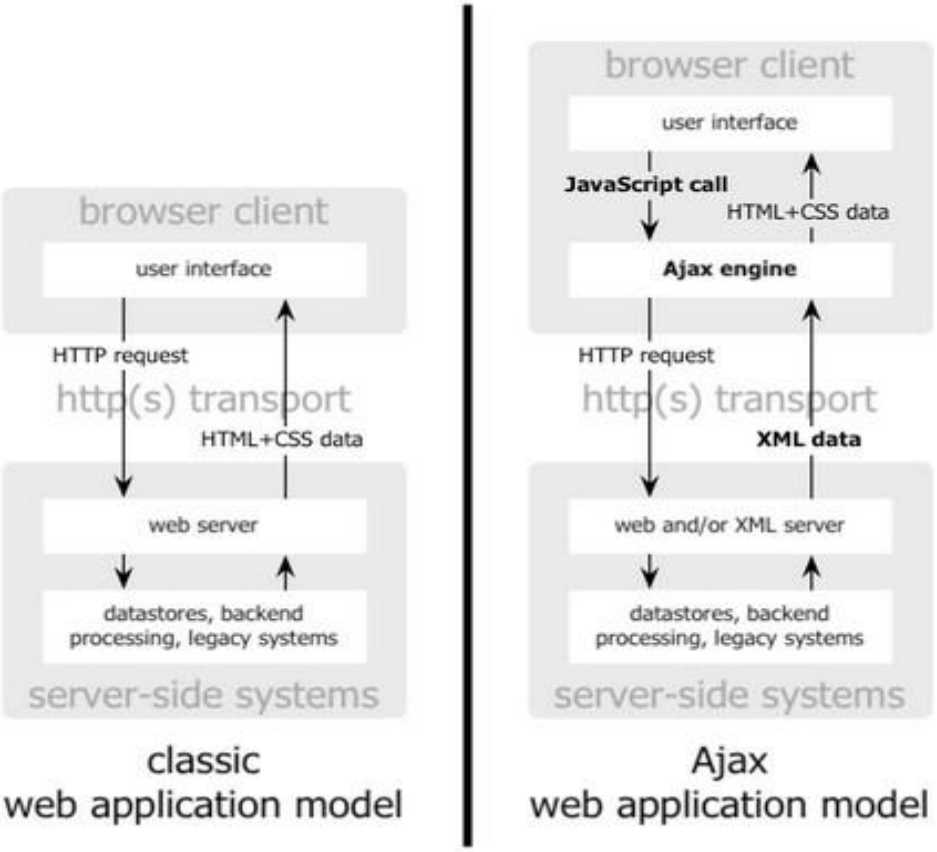
[1] Jedná se o objektově orientovaný programovací jazyk z dílny Sun Microsystems. http://www.linuxsoft.cz/article.php?id_article=268

1.2 Popis technologie AJAX

AJAX (Asynchronous JavaScript and XML) je ve své podstatě technologie, která slouží pro vývoje moderních internetových aplikací. Tato technologie umožňuje provádět interaktivní změny v aplikaci bez nutnosti opětovně načítat celkový obsah aplikace, jak tomu je u skriptovacích jazyků na straně serveru. Jak již z názvu vyplývá, jedná se o skupinu technologií, které zajišťují tuto komunikace od začátku až do konce. Ajaxové aplikace využívají takzvanou asynchronní komunikaci internetového prohlížeče se serverem. Tedy uživatel určí, které hodnoty chce zpracovat nebo zobrazit a server zpracovává pouze tento požadavek.

Tento způsob komunikace je realizován pomocí objektu XMLHttpRequest. Tento objekt je implementován ve všech grafických současných internetových prohlížečích a volán vždy pokud chceme vytvářet spojení se serverem. Podrobněji je popsán v následující podkapitole 1.3.

Technologie AJAX není samostatná technologie, ale jedná se o sloučení několika dostupných technologií. Spojení těchto technologií zajišťují interaktivní internetové aplikace bez nutnosti načítání celého obsahu při vyvolání události nebo provedení změny. Mezi tyto technologie patří značkovací jazyk využívaný pro popis struktury a obsahu internetové aplikace. Nazývá se (X)HTML a vychází ze spojení HyperText Markup Language a eXtensible Markup Language. S tímto jazykem je spjato stylování XHTML dokumentů takzvané CSS (Cascading Style Sheets), které slouží k vytváření vzhledu obsahu internetové aplikace. Další zásadní technologií je jazyk JavaScript. Ajaxové aplikace jsou napsány právě v tomto jazyce a umožňují tak využívat veškeré možnosti tohoto skriptovacího jazyka. Pro přístup k prvkům obsahu internetové aplikace a jeho modifikaci využívá JavaScript takzvaný standardizovaný Strom objektů DOM (Domument Object Model). A neméně důležitý je značkovací jazyk XML, který souží k samotné komunikaci serveru s aplikací postavené na technologii AJAX. Tato komunikace je nejčastěji realizována právě pomocí XML dokumentů. Existuje však ještě možnost získávat zprávy od serveru v podobě prostého textu.



Obrázek 1 - Klasická a Ajaxová aplikace [4]

Na obrázku 1 vidíme rozdíl mezi klasickou aplikací a aplikací psanou v AJAXu. V obou případech uživatelské rozhraní, tedy uživatel, vysílá požadavek na server pomocí protokolu HTTP (Hypertext Transfer Protocol). Tento protokol nabízí dvě metody pro přenos dat. Jedná se o metody GET a POST. Oba modely mohou využít kteroukoli metodu. Ale rozdíl v případě aplikace klasické a s využitím technologie AJAX je v tom, že Ajaxová aplikace tento požadavek realizuje pomocí objektu XMLHttpRequest, zatímco klasická aplikace přímo pomocí HTTP. Na obrázku je tento objekt znázorněn jako Ajax engine. Dále server provádí operaci na základě požadavku uživatele. Většinou se jedná o dotaz na databázový server nebo provedení nějakého skriptu na straně serveru. Informaci o výsledném stavu operace, tedy v případě databázového dotazu požadované informace získané přímo z databáze nebo v případě provedení skriptu o jeho úspěšnosti provedení, předá server zpět uživatelskému rozhraní. Pokud se ale jedná o Ajaxovou aplikaci, server výslednou informaci odešle pomocí XML dokumentu, jak je to znázorněno na obrázku 1. Tento dokument je zpracován v rámci XMLHttpRequest a pomocí technologie JavaScript

zobrazen v uživatelském rozhraní. Tento proces probíhá bez nutnosti obnovení celého obsahu načtené aplikace na rozdíl od klasické aplikace. Ta musí výslednou informaci zobrazit pomocí obnovení celého obsahu uživatelského rozhraní.

Moderní internetové prohlížeče mají v sobě zabudovanou možnost ukládat si načtené aplikace do vyrovnávací paměti, které se říká cache. Ta umožňuje po opětovném načtení stejných souborů načítat část obsahu právě z vyrovnávací paměti cache. Velkou nevýhodou je, že nemůžeme určit, které části se do paměti uloží a které ne. Může se stát, že po opětovné načtení obsahu souborů se načte měněná část dat z paměti prohlížeče na místo rovnou ze serveru. Dojde k tomu, že se nezobrazí aktuální data.

Jazyk PHP umožňuje předcházet tomuto problému pomocí funkce header. Headers neboli hlavičky jsou součástí protokolu HTTP a doprovázejí každý požadavek. PHP funkce header dokáže tyto hlavičky upravovat nebo naopak generovat pro konkrétní požadavek. Jednou z možností je právě odeslání hlavičky, která zakazuje ukládat aktuální obsah a tím zabrání uložení do paměti cache. V tabulce 1 jsou uvedeny další HTTP hlavičky.

Možnost ukládat konkrétní dokumenty do vyrovnávací paměti je běžnou součástí každého moderního internetového prohlížeče. V klasické aplikaci lze chování této paměti určitým způsobem ovlivnit. V aplikacích využívajících technologii AJAX se s touto skutečností nemusí vůbec počítat. Soubory aplikace se při prvním spuštění nahraje do vyrovnávací paměti prohlížeče a dále je už ovlivňována pouze ze strany klienta pomocí JavaScriptu. Stránka se již při posílání požadavku přes HTTP protokol znovu nenačítá, a proto není vyrovnávací paměť využívána. Mění se již pouze určitá část dat.

Tabulka 1 - HTTP hlavičky [3]

Název	Popis	Hodnota
Cache-Control	Ukládání do vyrovnávací paměti	Cache-Control: no-cache
Connection	Uzavření spojení	Connection: close
Pragma	Ukládání do vyrovnávací paměti	Pragma: no-cache
Location	Přesměrovaní dokumentu	Location: http://www.upce.cz/
Referer	URL, ze které uživatel přišel	http://www.upce.cz/
User-Agent	Identifikace prohlížeče	Mozilla/4.0
Server	Identifikace serveru	Server: Apache/1.4b2
Content-Type	Typ dokumentu	Content-Type: text/html
Expires	Platnost dokumentu (do té doby se neukládá do cache paměti)	Expires: Fri, 22 Jun 2001 17:18:17 GMT
Last-Modified	Poslední aktualizace obsahu	Fri, 22 Jun 2001 17:18:17 GMT

1.3 Objekt XMLHttpRequest

Objekt XMLHttpRequest (XHR) představuje jakési rozhraní pro skriptovací jazyky nejčastěji JavaScript. Umožňuje vysílat požadavky na server a přijímat odpovědi pomocí HTTP protokolu. Jak je z názvu patrné, odpovědi se nejčastěji realizují pomocí XML dokumentů, avšak není to jediná možnost. Odpověď může mít také podobu prostého textu. Tedy lze předávat části XHTML kódu nebo prostě jenom čistý text. V praxi se nejvíce využívá způsob s XML odpovědí.

Samotné spojení je realizováno pomocí HTTP protokolu. Existují dvě metody, které lze pro toto spojení využít. Jedná se o metody GET a POST. Metoda GET připojuje

zasílaná data přímo do URL serveru pomocí proměnných na rozdíl od metody POST, která data posílá v HTTP hlavičce skrytě pro uživatele. Data v proměnných jsou zakódována do podoby vyhovující URL a jejich výpis začíná za značkou ?.

http://upce.cz/index.php?promenna=1

Jedním z omezení spojení při využití tohoto objektu je, že HTTP požadavek může být poslán pouze v rámci stejné domény. Nelze tedy tímto způsobem vytvářet komunikaci mezi odlišnými servery. Toto omezení má bezpečností důvody. Pokud by to objekt umožňoval, potencionální útočník, by mohl získat neveřejné informace nebo podstrčit škodlivý kód.[9]

Jak jsem již zmínil v předchozí podkapitole, objekt XMLHttpRequest je implementován ve všech současných prohlížečích. Jediná odlišnost nastává při vytváření instance objektu. Rozdíl je u prohlížeče Microsoft Internet Explorer, dále jen MSIE.

Volání objektu pomocí MSIE.

var xmlhttp = new ActiveXObject("Microsoft.XMLHTTP");

Volání objektu v případě, že se nejedná o MSIE.

var xmlhttp = new XMLHttpRequest();

Pro bezpečné a univerzální volání objektu lze využít ternárních operátorů.

var xmlhttp = (window.XMLHttpRequest ? new XMLHttpRequest :
(window.ActiveXObject ? new ActiveXObject("Microsoft.XMLHTTP") : false));

Dále již se práce s objektem shoduje ve všech internetových prohlížečích. Nejdůležitější metodou tohoto objektu je metoda *Open*. Slouží k otevření spojení.

xmlhttp.open(method, url);

Parametr *method* slouží k určení metody HTTP požadavku, tedy metoda POST nebo GET. Druhý parametr *url* určuje adresu daného skriptu na serveru, který chceme volat. Pokud je spojení nastavené a otevřené zavolá se metoda *Send*. Tato metoda zajistí samotné odeslání dat na požadovanou URL adresu.

xmlhttp.send(content);

V parametru *content* se posílají data, která mají být odeslána zpravidla metodou POST. Jedná se tedy o data, která jsou připojena k požadavku na danou adresu *url*, ale nejsou její součástí. V případě, že se jedná o metodu GET, data jsou připojena k parametru adresy URL, stávají se její součástí. Objekt dále také umožňuje nastavit nebo zjistit hlavičky HTTP protokolu. K tomu slouží metody setRequestHeader a getRequestHeader popřípadě getAllResponseHeaders.

Nastavit hlavičky lze pomocí této metody.

xmlhttp.setRequestHeader(header, value);

Naopak zjistit hodnotu hlavičky umožňuje metoda.

xmlhttp.getResponseHeader(header);

Další metodou je *onreadystatechange*. Této metodě se přiřadí funkce, která zpracovává výsledný stav, tedy odpověď ze serveru. Metoda vychází z atributu *readyState*, který informuje, v jakém stavu se vyslaný požadavek nachází. Tento atribut může nabývat následujících hodnot, které jsou uvedeny v tabulce 2.

Tabulka 2 - Hodnoty atributu readystate [5]

Hodnota	**Popis stavu**
0	Vytvořena instance objektu
1	Zavolána a nastavena metoda Open
2	Zavolána metoda Send
3	Server odpovídá, ale odpověď není kompletní
4	Odpověď je kompletní

Pokud hodnota atributu je rovna 4, začneme zpracovávat odpověď a předávat jí do internetové aplikace pomocí DOM a JavaScriptu.

xmlhttp.readyState == 4

V případě, že by bylo z jakýchkoliv důvodů potřeba přerušit spojení vytvořené pomocí objektu XMLHttpRequest, využije se metoda *Abort*. Tato metoda zajistí zrušení spojení s okamžitou platností.

xmlhttp.Abort();

1.4 JavaScript a DOM

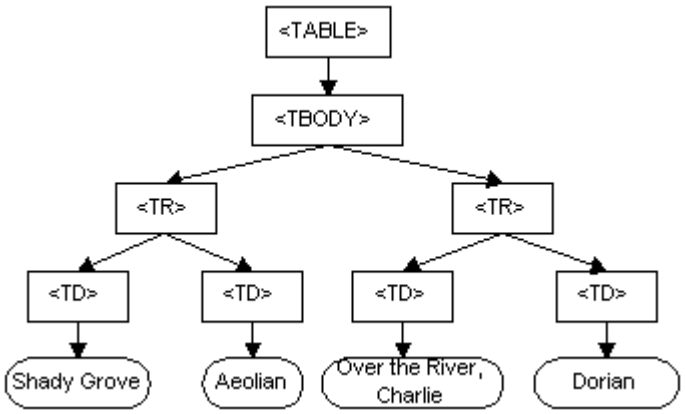
Při práci s objektem XMLHttpRequest se využívá skriptovací jazyk Javascript. Tento jazyk je nezávislý na platformě a objektově orientovaný. Jeho uplatnění je velmi široké, ale dominantní je při tvorbě internetových aplikacích. Jedná se o skriptovací jazyk na straně klienta, tedy pro svou práci využívá operační prostředky uživatelského počítače. Jeho největší nevýhodou je možnost tento jazyk zakázat v nastavení internetového prohlížeče. Jazyk se stává nefunkční a skripty v něm napsané nebudou spuštěny. Naopak jeho velikou výhodou je skutečnost, že nezatěžuje server. Jeho výkonnost se tak odvíjí od výkonu klientského počítače.

Jako spouštěcí mechanizmus funkcí pracujících s objektem XMLHttpRequest se využívají JavaScriptové události. Tyto události může uživatel vyvolat pomocí klávesnice, tlačítkem nebo kolečkem na myši, pohybem myší nebo změnou velikosti okna v (X)HTML stránce a tím spustit HTTP požadavky na server pomocí technologie AJAX. V následující tabulce je výčet nejpoužívanějších událostí, které lze při konstrukci využít.

Tabulka 3 - Události JavaScriptu [6]

Událost	Popis události
onLoad	při úplném načtení stránky
onUnload	při opouštění stránky
onResize	při změně velikosti okna
onScroll	při skrolování, posouvání obsahu
onClick	při kliknutí na prvek nebo při přednastavené akci
onDblClick	při dvojkliku na prvek
onMouseOver	při najetí myší na prvek
onMouseOut	při odjetí z prvku
onMouseMove	při pohybu myši nad prvkem
onMouseDown	při stisknutí tlačítka nad prvkem
onMouseUp	při uvolnění tlačítka nad prvkem
onKeyPress	při stisknutí klávesy ve chvíli, kdy je element aktivní
onKeyDown	při stlačení klávesy ve chvíli, kdy je element aktivní
onKeyUp	při uvolnění klávesy ve chvíli, kdy je element aktivní
onSubmit	těsně před odesláním formuláře
onFocus	při aktivaci políčka (okna)
onBlur	při deaktivaci políčka (okna)
onChange	při změně hodnoty políčka
onSelect	při vybrání textu myší
onAbort	při přerušení načítání

Pro přístup k objektům obsahu v internetové aplikaci využívá JavaScript objektový model dokumentu takzvaný DOM. „Technologie DOM je jedna z technologií, které umožňují pracovat s daty XML dokumentu. Jedná se o rozhraní nezávislé na konkrétní platformě ani programovacím jazyku, které vzniklo na půdě W3C konsorcia, a umožňuje aplikacím dynamicky přistupovat a aktualizovat obsah, strukturu a styl dokumentů. Primárně je tento standard určen pro práci s HTML a XML dokumenty a případně i dalšími typy dokumentů, které jsou založeny na podobně strukturovaném systému (např. WML, XHTML)[8]”.



Obrázek 2 - Ukázka DOM pro element TABLE [11]

Metody, které jsou ve skriptovacím jazyce Javascript implementovány pro práci s DOM, se využívají také při zpracování XML dokumentu. Jedná se zejména o metody *getElementsByTagName* a *getElementById*. Tyto metody zpřístupní element obsahu dokumentu na základě atributu id nebo na základě jména elementu. Poté lze elementy modifikovat jejich obsah nebo upravovat jejich vzhled s využitím CSS a JavaScriptu.

Odpověď ze serveru, která splňuje, že hlavička odpovědi *content-type* má hodnotu *text/xml* nebo *application/xml*, tedy, že se jedná o XML dokument, je převedena do DOM stromu a jeho obsah se nachází v *responseXML*. Pro konkrétní přístup k datům se využívá konstrukce.

```
var text = xmlhttp.responseXML.getElementsByTagName('text');
```

Kde proměnná text je pole, ve kterém jsou všechny výskyty elementu text. Pokud bychom chtěli tento obsah předat nějakému elementu internetové aplikace uživatelského rozhraní, využili bychom tuto konstrukci.

```
var text_element = document.getElementById('text_element');
var text = xmlhttp.responseXML.getElementsByTagName('text');

for (var i=0; i < text.length; i++){
        text_element.innerHTML+= text[i].firstChild.data;
}
```

Pomocí cyklu se naplnil element *text_element* obsahem z odpovědi serveru. Zpracování odpovědi zařizuje funkce, která je volána po každé změně stavu požadavku. Je proto nutné zjistit v těle této funkce, jestli je v atributu *readyState* nastavena hodnota na číslo 4, tedy jestli je požadavek kompletně vyřízen.

```
If(xmlhttp.readyState == 4){
        //zpracování odpovědi
}
```

Pokud by hodnota byla jiná, nepodařilo by se přistoupit k datům odpovědi pomocí responseXML. V takovém případě by funkce vrátila hodnotu NULL.

2. AJAX a jeho využití

Technologie AJAX je bezesporu technologie posouvající hranice v tvorbě interaktivních internetových aplikací. Její možnosti dovolují vytvářet uživatelsky přívětivější internetové aplikace. Umožňují velmi snadno ovládat, spravovat nebo vytvářet obsah XHTML stránek. Může se zdát, že technologie AJAX je velmi výkonný a bezchybný nástroj, který se dá aplikovat na jakýkoliv projekt. Bohužel i tato technologie má spoustu nevýhod, které se musí brát v úvahu, při rozhodování, zda využít technologii AJAX nebo se rozhodnout pro klasickou variantu řešení. Především je důležité si ujasnit výhody a nevýhody této technologie a poté zvážit je-li nasazení technologie AJAX efektivní.

2.1 Výhody a nevýhody technologie AJAX

Zásadní výhodou technologie AJAX je zrychlení internetové aplikace, která se nemusí po každém požadavku na server ze strany klienta neustále celá znovu načítat. To je vlastnost, která přibližuje internetové a intranetové aplikace ke klasickým takzvaným desktopovým aplikacím. Uživateli to umožňuje rychlé a komfortní ovládání aplikace. Na server odcházejí požadavky pouze části aplikace a odpověď ze strany serveru je ve velmi krátkém čase zobrazena zase pouze aktualizací nějakého prvku v XHTML stránce. Uživatel tedy dostává pocit plynulé a snadné práce, jako by pracoval s klasickou aplikací ve svém operačním systému. [9]

Naopak nevýhod, které tato technologie při nasazení přináší, je celá řada. Je to dáno hlavně tím, že technologie AJAX je složena z několika dostupných technologií. Ty s sebou nesou také určité nevýhody. Jednou z takových nevýhod je skutečnost, že programovací jazyk JavaScript lze v internetovém prohlížeči vypnout. Ať už z bezpečnostních důvodů nebo jenom z neznalosti. V takovém případě technologie AJAX, která je postavena na JavaScriptu, nebude fungovat vůbec. Pro tyto případy se musí vytvořit alternativní větev aplikace bez využití technologie AJAX. [9]

Další nevýhodou je implementace samotného AJAXU v internetových prohlížečích. V kapitole 1.3 je uvedena rozdílná inicializace stejného objektu XMLHttpRequest pro dva dominantní internetové prohlížeče. Což může vést při dnešním rychlém

vývoji nových verzí prohlížečů k nekompatibilitě se staršími verzemi při inicializaci objektu XMLHttpRequest.

Jednou z dalších nevýhod je problém při používání tlačítka Zpět v internetovém prohlížeči, které slouží pro návrat na předchozí navštívenou stránku a je součástí každého internetového prohlížeče. Používání tlačítka Zpět je velmi rozšířené a stalo se základní zvyklostí každého uživatele internetu. Bohužel technologie AJAX tento zvyk nabourává. Většinou tlačítko zpět nefunguje. Pokud však funguje, vrací na stránku před tím, než se využila nějaká komunikaci prostřednictvím technologie AJAX. V podstatě se nedokáže vracet po změnách obsahu stránky, které byly změněny technologií AJAX.

Další nevýhoda vychází z podstaty koncepce technologie AJAX. S nárůstem kódů napsaných v jazyce JavaScript se přesunulo určité množství zátěže na internetový prohlížeč a tím tedy na samotný počítač uživatele. Problém s výkonností samotné aplikace by mohl nastat v případě, že uživatel bude mít spuštěno více aplikací postavených na technologii AJAX. V takovém případě by se průběh požadavků z hlediska času značně zvýšil. Je však nutné říci, že dnešní velmi výkonné počítače by s tímto ojedinělým stavem neměly mít vůbec problém. Týká se to spíše starších méně výkonných počítačů. [9]

2.2 Nejčastější použití technologie AJAX

V souvislosti s výhodami a nevýhodami této technologie existují oblasti, které jsou ideální pro nasazení technologie AJAX. Mezi prvními vývojáři, kteří se rozhodli využívat AJAX pro zlepšení komfortnosti práce uživatelů s jejich prostředím, je firma Google Inc. Jedná se především o takzvaný našeptávač, který po zadávání textu do pole pro vyhledávání podsouvá, našeptává nejčastěji hledané fráze ostatních uživatelů a počet hledání těchto frází. Následující obrázek představuje našeptávač vyhledávače Google.cz.

Obrázek 3 - Našeptávač vyhledávače Google.cz[vlastní]

Je to jistě velmi užitečná pomůcka při formulaci hledané fráze. Dnes se tento doplněk rozšířil do všech současných vyhledavačů a využívá se i mimo ně. Jeho uplatnění lze nalézt v různých dalších oblastech. Například při zadávání nebo úpravách dat ve formulářích.

Našeptávač využívá JavaScriptovou událost *onChange*, která je přiřazena formulářovému prvku pro zadání textu pro vyhledávání. V případě, že se vyvolá tato událost, zavolá se funkce, která obsluhuje XMLHttpRequest objekt. Pomocí metod tohoto objektu se vytvoří a pošle požadavek na databázový server, který vrátí přesné nebo podobné fráze textu, který byl zadán do formulářového pole. S využitím technologie DOM a JavaScriptu se výsledný seznam zobrazí v prvku XHTML stránky, který je umístěn přesně pod vyhledávacím pole. Tento proces se opakuje vždy, když uživatel změní obsah pole pro zadání textu. Pro vyvolání tohoto procesu lze také využít události zachycující stisknutí kláves *onKeyPress*. Popis těchto událostí se nachází v tabulce 3.

V dnešní době různých sociálních sítí a komunikátorů prostřednictvím internetu se klade důraz na různé způsoby diskuze. Jedním z těchto způsobů je takzvaný chat. Jedná se o diskuzi mezi uživateli. Ihned po vložení příspěvku jednoho uživatele se

zobrazí příspěvek všem bez nutnosti znovu načítat celou stránku. V dřívějších dobách se tento způsob řešil pomocí IFRAME[3]. Jednalo se o HTML značku, která umožňovala obnovení pouze jejího obsahu. Tato značka však měla velké nevýhody a později se doporučovalo ji nevyužívat. Ideálním řešením pro tuto problematiku je technologie AJAX. Pole pro diskuzi může být umístěné kdekoliv na stránce. AJAX zajistí vkládání příspěvků a zobrazování příspěvků na základě nastaveného časové intervalu. Časový interval zajišťují JavaScriptové funkce *setInterval* a *clearInterval*. Nevýhoda je zřejmá. Zobrazení nových příspěvků neprobíhá v čase jejich vložení, ale v čase pravidelného obnovení a při snížení intervalu roste náročnost aplikace na straně klienta.

Další využití technologie AJAX je při registraci nových uživatelů nebo při zadávání a ověřování dat do databáze. V případě registrace je v některých případech nutné ověřit, zdali uživatel se zadaným emailem již neexistuje, tedy jestli již není uložen v databázi. Ověření pomocí technologie AJAX je velmi rychlé a interaktivní. Toto ověřování funguje stejným principem jako našeptávač. Jediný rozdíl je, že server vrací pouze informaci o tom, je-li zadaná emailová adresa volná nebo obsazená. V tomto případě je nutné počítat s tím, že uživatel může mít JavaScript vypnutý nebo zakázaný a je zapotřebí vytvořit alternativní kontrolu na straně serveru.

2.3 Zhodnocení technologie AJAX z hlediska výkonu

Výkonnost internetových aplikací je velmi důležité kritérium při posuzování aplikace jako celku. Především na internetu, kde je samotná výkonnost aplikace ovlivňována několika faktory. Nejvýznamnějším faktorem, je výkon samotného serveru, na kterém je aplikace umístěna. V případě, že aplikace využívá nějaký databázový server, který je nainstalován společně s webovým serverem na jednom serverovém počítači, nároky na výkonnost serveru vzrostou. Dalším faktorem je samotný návrh a konstrukce internetové aplikace z hlediska programování. Kvalitním navržením databáze se výrazně sníží zátěž serveru a tím zvýší výkonnost celé aplikace. Jedná se především o vhodné navržení indexů sloupců v jednotlivých tabulkách. [10] Výkonnost lze ovlivnit také dobře napsanými zdrojovými kódy aplikace, vyhnout se zbytečným cyklům nebo používat perzistentní připojení k databázím[4].

[3] http://www.jakpsatweb.cz/html/ramy.html
[4] http://php.vrana.cz/persistentni-pripojeni.php

Ovšem zásadním faktorem, který ovlivňuje výkonnost aplikace na internetu, jsou možnosti HTTP protokolu. Jak již bylo řečeno v předchozích kapitolách, pomocí HTTP protokolu aplikace komunikuje se serverem. Při každém požadavku se aplikace načítá celá, ačkoliv se aktualizuje pouze jedna část. V tomto případě se nabízí využití technologie AJAX, která tuto problematiku řeší pomocí asynchronní komunikace se serverem přes protokol HTTP. Výkon aplikace by se tedy měl zvýšit, protože se ze serveru nestahují všechna data, ale jen ta, která se aktualizují.

V následující tabulce 4 je porovnání výkonu skutečné aplikace, která v jednom případě využívá technologii AJAX a v druhém případě klasický přístup. Aplikace funguje jako jednoduchý chat mezi uživateli.

Pro eliminaci vnějších vlivů je měření rychlosti vykonávaných skriptů prováděno na lokálním serveru Apache 2.0.63 na počítači Dell Optiplex 360 s procesorem Intel^{R} Core^{TM}2 Duo 3GHz s operační pamětí 3 GB RAM a operačním systémem Microsoft Windows XP Professional. Pro samotné měření technologie AJAX je využito rozšíření internetového prohlížeče Mozilla Firefox 3.6.3 s názvem FireBug 1.5.3. Měření klasického přístupu probíhalo pomocí PHP funkce *microtime* s parametrem 1. Tato funkce vrací aktuální čas ve formátu "sekundy mikrosekundy" a pokud se uvede parametr s hodnotou 1, vypíší se pouze sekundy. Na začátek skriptu se umístila proměnná s hodnotou vrácenou funkcí *microtime* a na konci se do jiné proměnné umístila nová hodnota funkce *microtime*. Výsledný čas se získal rozdílem obou proměnných a následně se převedl na milisekundy.

Měřený skript vykonává vkládání příspěvky uživatelů do databáze. Pro objektivnější výsledek bylo měření prováděno pětkrát a výsledek byl určen z průměru všech pěti měření.

Tabulka 4 - Naměřené údaje vykonávaných skriptů v milisekundách[vlastní]

Číslo měření	**PHP**	**AJAX**
1.	7,53	8,01
2.	9,15	7,31
3.	8,36	7,03
4.	7,35	7,69
5.	9,2	8,03
průměr	**8,32**	**7,61**

Z tabulky 4 vyplývá, že pro tento konkrétní problém se v průměru o 0,71 milisekund jako výkonnější ukázalo použití technologie AJAX. Zajímavé však je, že to není jednoznačný výsledek. Při prvním měření vykonání skriptu s využitím asynchronní komunikace byla naměřená doba vyšší než u klasického způsobu pomocí PHP skriptu. Ve čtvrtém měření byl rozdíl velmi malý. V tomto případě tedy lze říci, že nasazení technologie AJAX je sice výkonnější, ale není hlavní výhodou tohoto využití. Jedná se hlavně o komfort uživatele, který chat využívá pro komunikaci s ostatními uživateli. Pomocí technologie AJAX se velmi výrazně zvyšuje uživatelská přívětivost a plynulost této aplikace.

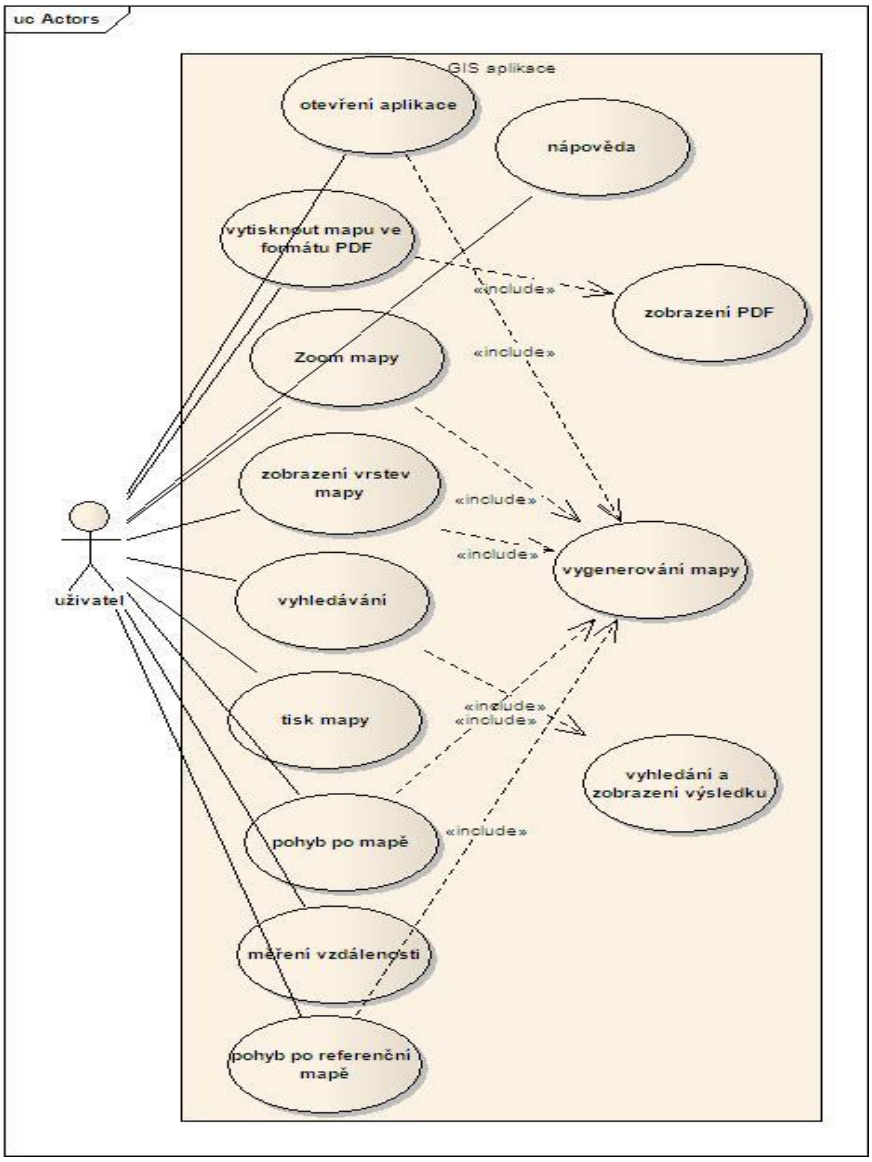
3 Návrh a vytvoření webového rozhraní GIS aplikace

Každý vývoj internetové aplikace začíná analýzou daného problému a tvorbou návrhového vzoru. Bez těchto procesů by samotná realizace nemohla začít. Je to součástí životního cyklu každého vývoje programových aplikací ať už internetových nebo desktopových.

3.1 Analýza a návrh

Tvorba návrhu GIS aplikace s využitím technologii AJAX vychází z diplomové práce Bc. Kamila Jakoubka s názvem **Návrh použitelného uživatelského rozhraní webového geografického informačního systému**. Tato práce se zabývá vhodným a efektivním návrhem uživatelského rozhraní GIS aplikace na základě dotazníkového šetření. Autor tyto údaje zpracovává a vyhodnocuje a vytváří nejlepší možnou variantu návrhu z hlediska pohledu práce uživatele s aplikací. Tato varianta se stala základním návrhovým vzorem pro další rozpracování a realizaci návrhu.

Nejprve je zapotřebí vytvořit model, který by zachycoval všechny interakce mezi uživatelem a samotnou aplikací. Nejlepším řešením je využití modelovacího jazyka UML (Unified Modeling Language). "Modelovací jazyk UML je souhrnem především grafických notací k vyjádření analytických a návrhových modelů[7]". Jednou z jeho součástí je Use case diagram. Tento diagram představuje komunikaci uživatele se systémem. Vymezuje přesně funkčnost dané aplikace. Na obrázku 3 je vytvořen Use case diagram navrhované GIS aplikace. Na základě tohoto diagramu se začal vytvářet objektový model, který byl realizován pomocí JavaScriptu a objektu XMLHttpRequest.



Obrázek 4 - Use case diagram GIS aplikace[vlastní]

Jednotlivé případy užití reprezentují jednotlivé funkce uživatelského rozhraní GIS aplikace a jsou v diagramu znázorněny elipsou a názvem případu užití. Actor s názvem uživatel reprezentuje uživatele obsluhujícího aplikaci. Přímé vztahy mezi uživatelem a případy užití jsou znázorněny plnou čarou. Vztah, který vyžaduje další případ užití je znázorněn čerchovanou čarou a relací s názvem include. Tato relace představuje závislost jednoho případu užití na druhém. Na základě tohoto diagramu se následně vytvořil funkční model, který se poté realizoval.

3.2 Funkčnost aplikace z pohledu uživatele

Z obrázku 3 vyplývá, že při otevření aplikace uživatelem se automaticky vygeneruje mapa. Toto vygenerování probíhá na základě výchozích hodnot a zobrazuje mapu ve výchozí pozici.

Pokud bude uživatel chtít vygenerovat mapu ve formátu PDF, klikne na možnost export do PDF. Systém následně převede aktuální mapu do formátu PDF a zobrazí ji v prohlížeči. Uživatel poté může výsledný PDF soubor uložit do svého počítače nebo ho popřípadě přímo vytisknout.

Dále může uživatel oddalovat nebo přibližovat mapu pomocí kolečka na myši, popřípadě posuvníku, který je umístěn v pravém horním rohu mapy pod tlačítky pro pohyb po mapě nebo pomocí tlačítka zoom na obrázku 4. Pohyb po mapě je realizován do čtyř poloh. Nahoru, dolů, doleva a doprava. Další možností jak se po mapě pohybovat je pomocí kurzoru myši. Uživatel klikne na mapu v místech, kam se chce posunout a aplikace mapu vygeneruje podle požadavků. Stejným způsobem se aplikace chová v případě funkce zoom.

Uživatel také může zobrazit a skrýt vrstvy, které jsou na mapě použity. Jedná se o vrstvy Kraje, Lesy, Vodní toky, Vodní plochy, Sídla, Železnice, Železniční stanice a Silnice. Pomocí formulářových prvků v menu Výběrem vrstvy může uživatel kombinovat viditelnost jednotlivých vrstev a tříd. V aplikaci je možné ovlivňovat viditelnost tříd u vrstvy Silnice, Železniční stanice, Vodní plochy a Vodní toky. Pokud je vrstva označena za viditelnou, uživatel může její třídy libovolně zobrazovat a skrývat. V opačném případě je vrstva skryta celá bez ohledu na nastavení tříd.

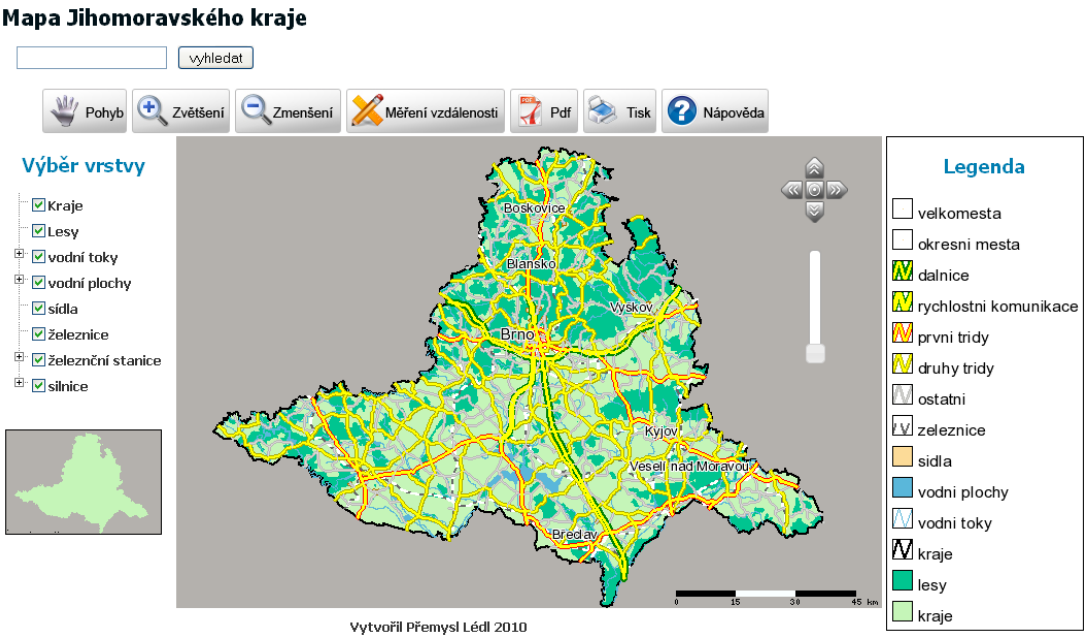
Vyhledávání objektů na mapě probíhá pomocí názvů objektu, které jsou tvořeny body. Jedná se o vrstvu sídel a vrstvu železničních stanic. Vyhledávání umožňuje zadávat hledaný text také bez velkých písmen a bez české diakritiky. Systém je nastaven tak, že zobrazuje kromě přesných výsledků také výsledky podobné začínající zadaným textem. Seznam výsledků se zobrazí v pravé části obrazovky. Po kliknutí na položku v seznamu se zobrazí položka na mapě.

Tisk výsledné mapy lze také realizovat rovnou bez nutnosti převádět mapu do formátu PDF. Tato možnost, kterou obstarává funkce tisk mapy, vytiskne všechny informace zobrazené na obrazovce včetně legendy, referenční mapy nebo menu pro výběr vrstev.

Velmi užitečná funkce pro uživatele je také měření vzdálenosti na mapě. Tato možnost se spouští tlačítkem měření vzdálenosti a umožňuje postupné kontinuální měření vzdálenosti. Po kliknutí na mapu se vytvoří počáteční bod. Každým dalším kliknutím myší na mapu se vzdálenost přičítá a zároveň se zobrazuje dráha na mapě pomocí červených úseček, které mají počáteční bod vždy v posledním bodě předešlé úsečky, popřípadě počátečního bodu měření. Zároveň s úsečkami se zobrazuje titulek, který informuje o celkové naměřené vzdálenosti v kilometrech.

Pro přehlednou a snadnou práci s mapou je v levém dolním rohu umístěna přehledová mapa, jak je možno vidět na obrázku 4. Tato mapa zobrazuje, jakou má uživatel aktuální pozici na mapě a jak velká část mapy je zobrazena. Její další užitečnou funkcí je možnost rychlého přesunu po mapě pomocí kliknutí myši na místo v referenční mapě, kam se chce uživatel přesunout.

Pokud uživatel nepochopí význam některých funkcí, aplikace mu nabízí stručnou nápovědu. V této nápovědě se uživatel dočte význam jednotlivých tlačítek reprezentujících základní funkce aplikace.



Obrázek 5 - Náhled GIS aplikace[vlastní]

3.3 Technická specifikace GIS aplikace

Celá aplikace je postavena na ovládání mapového serveru, který se jmenuje MapServer dříve University Of Minnesota MapServer[5]. Jedná se o Open source program, který je vyvíjen na univerzitě Minesota ve spojených státech amerických. MapServer lze využít a ovládat jako CGI (Common Gateway Interface) aplikaci nebo pomocí MapScript rozhraní. MapScript rozhraní existuje v různých programovacích jazycích například C#, JAVA nebo PHP. [1]

MapScript zajišťuje komunikaci mezi CGI aplikací a internetovou aplikací. Navzájem si předávají parametry pro vygenerování mapového souboru. Pro tvorbu GIS aplikace bylo použito MapScript rozhraní pro jazyk PHP.

$map_path="/var/www-stud/e078877/mapservdata/";

$map = ms_newMapObj($map_path."kraje.map");

Základním objektem v PHP MapScript, je objekt *ms_newMapObj*. Má jediný parametr odkazující na konfigurační soubor MapFile, který bude popsán na závěr této podkapitoly. Tento objekt obsahuje spoustu metod pro úpravu a editaci výsledné mapy na základě zaslaných parametrů. Jednou z nejdůležitějších je metoda *draw*. Tato metoda upravenou výslednou mapu vykreslí a pomocí metody *saveWebImage()* uloží na disk. Metoda vrátí URL vygenerované mapy.

$image=$map->draw();

$image_url=$image->saveWebImage();

MapScript následně vygeneruje XML soubor, který obsahuje informace o vygenerované mapě. MapScript je volán pomocí technologie AJAX, která jako odpověď využívá komunikaci pomocí XML souboru. Struktura XML souboru je následující.

[5] http://mapserver.org/

```
<image>
        <url>$image_url</url>
        <url_ref>$image_ref</url_ref>
        <url_scale>$image_scale</url_scale>
        <url_legend>$image_legend</url_legend>
        <extent_to_html>$extent_to_html</extent_to_html>
        <center_x>$x</center_x>
        <center_y>$y</center_y>
        <state>0</state>
</image>
```

Kořenovým elementem je element *image*. Tento element obsahuje již elementy listové, tedy elementy, které neobsahují žádné potomky. Elementy nesou informace o URL vygenerované mapy, referenční mapy, měřítka a legendy. Dále v elementu *extent_to_html* je uložena informace o aktuálních hraničních souřadnicích vygenerované mapy. Střed mapy reprezentují hodnoty elementů *center_x a center_y*. Element *state* uchovává hodnotu, v jakém stavu se mapa nachází. Pokud je mapa ve výchozí pozici, hodnota je rovna 0. V případě, že mapa je přiblížená na hraniční možnost, hodnota elementu *state* je rovna 5.

Další důležitou složkou je soubor, který se jmenuje MapFile. Tento soubor je velmi důležitý a nezbytný. Představuje jakýsi konfigurační soubor mapových dat. Jeho součástí je základní nastavení mapy. Velikost mapy, jednotky, cesta ke zdrojovým datům uloženým ve formátu shapefile[6], zobrazení mapy především elipsoid, dále také formát, ve kterém se mapa bude generovat, hraniční souřadnice. Dále se zde nastavují vlastnosti jednotlivých vrstev a jejich tříd. [2]

Následující ukázka představuje nastavení vrstvy vodní plochy.

[6] Jedná se o formát vektorových dat od společnosti ESRI.
http://projekty.geolab.cz/gacr/a/files/cajt_fmadpz05.pdf

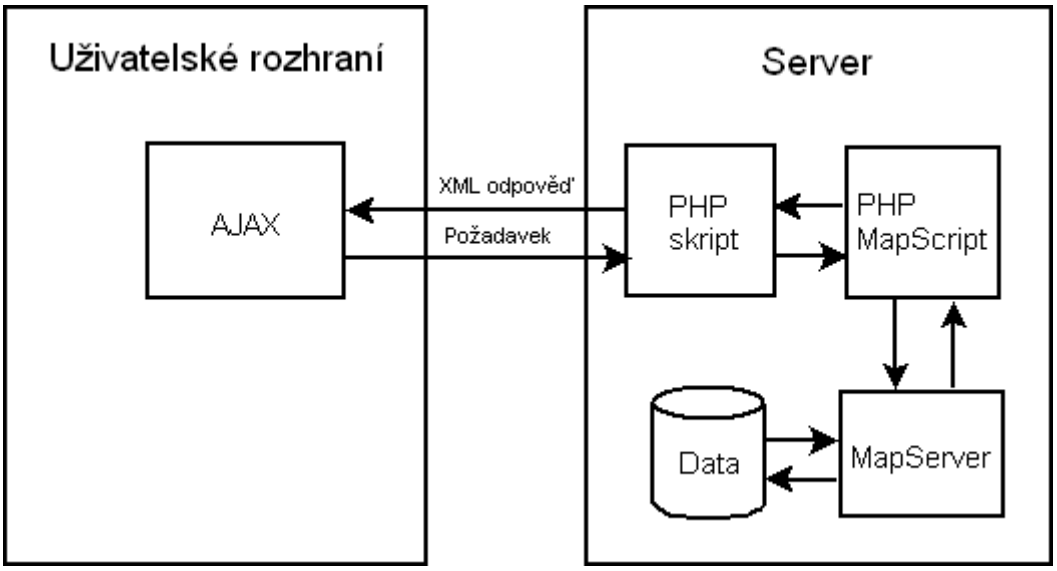
```
LAYER
   NAME       vod_pl
   DATA       vod_pl
   TYPE       polygon
   STATUS     ON


   CLASS
      NAME 'vodni plochy'
      LABEL
                  ENCODING Windows-1250
                  COLOR  90 183 250
                  OUTLINECOLOR 255 255 255
                  FONT arial
                  TYPE truetype
                  SIZE 8
                  POSITION AUTO
                  PARTIALS FALSE
      END
      STYLE
         COLOR 90 183 218
      END
   END
END
```

V první části se nastavují základní údaje. Jméno vrstvy, jméno vrstvy uložené v shapefile. Dále také typ vrstvy v tomto případě polygon, protože se jedná o vodní plochu. V další části se nastavují třídy. Třídy se většinou vytvářejí na základě nějakého atributu vrstvy. Například rozdělení podle rozlohy. Každé třídě lze nadefinovat různý vzhled, jméno, popisek nebo symbol. Tyto vlastnosti lze měnit dynamicky za běhu aplikace pomocí PHP MapScript.

Základní komunikaci a funkčnost obstarává technologie AJAX. Při každém požadavku uživatele na aplikaci technologie AJAX volá pomocí objektu XMLHttpRequest PHP skript *aplikace.php* s parametry požadavku pomocí HTTP metody GET. Dále očekává odpověď v podobě XML. Na obrázku 7 je tato komunikace zobrazena jako XML odpověď, kterou vygeneruje PHP skript. Pomocí technologie DOM a JavaScript zobrazí získané informace o obsahu aplikace. V případě uživatelského požadavku vyhledat data na základě zadaného textu, AJAX volá PHP skript *query.php*. Tento skript ve své XML odpovědi vrací všechny vyhovující výsledky s příslušným označením vrstvy, do které přísluší. AJAX tento

výsledek převede do seznamu s odkazy na konkrétní místa na mapě a zobrazí místo legendy.

Na obrázku 7 je znázorněna komunikace mezi serverem a klientským počítačem webového rozhraní GIS aplikace. Uživatel pomocí AJAXu vyšle požadavek na server. PHP skript zpracuje parametry požadavku a prostřednictvím PHP MapScript je předá MapServeru. Data představují soubory shapefiles a soubor MapFile. Tyto soubory využívá MapServer, který na základě parametrů a dat vygeneruje požadovanou mapu. Informace o výsledné vygenerované mapě PHP skript zpracuje do XML odpovědi, kterou očekává AJAX. AJAX XML odpověď zobrazí pomocí DOM a JavaScriptu uživateli.



Obrázek 6 – Komunikace mezi serverem a uživatelským rozhraním[vlastní]

3.4 Posouzení vhodnosti využité technologie AJAX

Jedním z cílů této bakalářské práce je posoudit efektivnost zvolených technologií pro tvorbu GIS aplikace. Rozhodnout na základě stanovených kritérií, zda využití technologie AJAX je tou správnou volbou, popřípadě porovnat výhody a nevýhody nasazení pro tuto GIS aplikaci.

Jedním z kritérií, které jsou zásadní pro vyhodnocování a porovnávání, je rychlost vykonávaných skriptů. Rychlost přímo souvisí se samotným výkonem celé aplikace.

Je závislá na hardwarových prostředcích serveru nebo na rychlosti připojení uživatele, který pracuje s aplikací. V současnosti jsou však servery velmi výkonné a rychlost připojení je dostačující na to, aby rychlost aplikace byla ovlivněna z větší části samotnou konstrukcí a zvolenými technologiemi.

Pro objektivní zhodnocení využití technologie AJAX je zapotřebí vytvořit GIS aplikaci pomocí klasického přístupu. Tato testovací aplikace je napsána ve skriptovacím jazyce PHP a komunikace se serverem probíhá na základě HTTP protokolu pomocí metody GET. Důležité je, že komunikace pomocí HTTP protokolu probíhá synchronně. Funkčnost testovací aplikace je stejná jako funkčnost aplikace využívající technologii AJAX, která využívá asynchronní komunikaci přes HTTP protokol. Rozdíl mezi oběma aplikacemi je v tom, že testovací aplikace při každé odpovědi ze serveru se načítá celá i s částmi, které nebyly aktualizovány. Aplikace vytvořená pomocí technologie AJAX aktualizuje pouze části, které se na základě zaslaného požadavku na server změnily.

Měření rychlosti vykonávaných skriptů obou aplikací proběhlo na vybraných funkcích aplikací, které reprezentují nejdůležitější oblasti možností aplikace. Jedná se o pohyb po mapě pomocí posuvníku. Dále je to výběr vrstvy, přiblížení mapy a poslední je funkce, která umožňuje přechod do výchozího stavu mapy. Tato funkce se vyvolá pomocí tlačítka umístěného uprostřed posuvníku. Samotné měření probíhalo u obou aplikací pomocí jiného postupu. U testovací aplikace se využila PHP funkce *microtime*. Podobně jako u předchozího měření doby probíhajících skriptů. Rozdíl naměřených hodnot byl pomocí PHP funkce *echo* zobrazen na konci skriptu a zobrazený výsledek představoval dobu potřebnou na vykonání skriptu v sekundách. Pro naměření hodnot aplikace postavené na technologie AJAX bylo využito rozšíření internetového prohlížeče Mozilla Firefox 3.6.3 s názvem FireBug 1.5.3., které umožňuje sledovat a vyhodnocovat dobu načítání skriptů využívající technologii AJAX. Výsledky měření byly převedeny na milisekundy a jsou uvedeny v následující tabulce.

Tabulka 5 - Doba vykonávaných skriptů GIS aplikace v milisekundách[vlastní]

	PHP	**AJAX**
pohyb po mapě	202	271
výběr vrstvy	375	198
přiblížení na mapě	180	193
přechod do výchozího stavu	294	241
Průměr	**263**	**226**

Z dílčích naměřených hodnot jednotlivých funkcí byl vytvořen průměr, který představuje výslednou hodnotu pro zvolenou technologii. Z výsledných čísel má menší hodnotu průměr, který reprezentuje technologii AJAX. Na základě vytvořeného průměru lze říci, že technologie AJAX v celkovém kontextu GIS aplikace je rychlejší, má menší dobu pro vykonání základních funkcí aplikace. Rozdíl obou průměrů je roven 37 milisekund. To je velmi krátká doba a pro běžného uživatele, který pracuje s aplikací, zanedbatelná.

Pokud se však porovnají naměřené výsledky jednotlivých funkcí pro každou funkci zvlášť, zjistí se, že doba u aplikace s technologií AJAX není vždy menší. Dokonce v jednom případě přesahuje testovací aplikaci o více, než je rozdíl obou průměrů. Ve druhém případě je přesah menší, ale stále je doba vykovávaných skriptů delší, než u testovací aplikace napsané v jazyce PHP. Pro funkce pohyb po mapě a přiblížení mapy dopadla aplikace s technologií AJAX hůře, než testovací aplikace.

Naproti tomu testovací aplikace měla velké problémy s vykonáním funkce výběr vrstvy. Zde je rozdíl časů nejmarkantnější a je to hlavní důvod, proč v celkovém hodnocení dopadla lépe aplikace s technologií AJAX. Další problémovou funkcí pro testovací aplikaci je přechod do výchozího stavu. Rozdíl měření u obou aplikací ukázal, že testovací aplikace přesáhla o 53 milisekund aplikaci využívající technologii AJAX. Tyto dvě funkce jsou pro testovací aplikaci výrazně časově

náročnější na zpracování než pro aplikaci s technologií AJAX a rozhodly o celkově horším umístění testovací aplikace v době potřebné pro vykonání skriptů.

Na základě výsledků testu měření doby potřebné pro vykonání skriptů v GIS aplikaci lze říci, že z hlediska rychlosti aplikace se jeví využití technologie AJAX jako efektivnější než v případě klasického přístupu. Ačkoli doby u všech jednotlivých funkcí nebyly vždy menší, v celkovém hodnocení pomocí průměru dílčích měření dopadla lépe aplikace využívající technologii AJAX. Ovšem její přednost spočívá také v komfortu a pohodlí, které uživateli aplikace poskytuje. Vytváří dojem plynulosti a umožňuje ve spojení s JavaScriptem rozšířené možnosti ovládání, které testovací aplikace ze své podstaty umožňovat nemůže.

Závěr

Jedním s cílů této bakalářské práce bylo představit technologii AJAX v kontextu rozvoje internetových aplikací a požadavků uživatelů internetu. V práci je popsána historie vzniku technologie AJAX a současný trend internetových aplikací. Představuje alternativní možnosti k technologii AJAX a popisuje jejich výhody a nevýhody. Především Java applett a technologie FLASH.

Dále jsou popsány technologie, které technologie AJAX využívá. Jedná se především o skriptovací jazyk JavaScript, ve kterém jsou AJAX skripty napsány. Dále značkovací jazyk XML a technologie DOM. Práce ukazuje jednotlivé funkce těchto technologií při komunikaci se serverem pomocí asynchronní komunikace HTTP protokolu a vysvětluje jejich úlohu v celkové komunikaci.

Základním a nejdůležitějším objektem technologie AJAX je objekt XMLHttpRequest. Práce ukázala, že jedním z hlavních problémů při implementaci tohoto objektu je rozdílná deklarace v internetových prohlížečích, které jsou v současnosti dominantní na internetovém poli. Dále popsala funkce a metody, které tento objekt umožňuje a jednotlivé parametry těchto funkcí.

Výhody a nevýhody technologie AJAX usnadňují pochopit, které oblasti nasazení v internetových aplikacích jsou vhodné nebo nezbytné, a naopak které jsou nevhodné. Mezi nejčastější řešené problémy u aplikací, které využívají technologii AJAX, je nefunkčnost tlačítka zpět v internetovém prohlížeči.

Pro přesné posouzení rozdílné náročnosti na výkon internetové aplikace při využití technologie AJAX nebo nasazení klasického způsobu komunikace se serverem byla vytvořena testovací aplikace. Tato aplikace představovala uživatelský chat. Test prokázal menší časovou náročnost u aplikace, která využívá technologii AJAX. Na základě těchto výsledků lze přesvědčivě říci, že uživatelský chat je vhodnou oblastí pro nasazení technologie AJAX.

Hlavním cílem práce bylo vytvořit GIS aplikaci využívající technologii AJAX a posoudit vhodnost a efektivitu zvolené technologie. Pro vytvořenou aplikaci byla vytvořena testovací aplikace, která byla napsána pomocí jazyka PHP. Obě aplikace byly otestovány na vybraných funkcích aplikace. V celkovém hodnocení dopadla lépe aplikace využívající technologii AJAX.

Na základě obou testů by se dalo říci, že technologie AJAX klade menší nároky na server a tím dosahuje lepší rychlosti a plynulosti běhu aplikace. Je také nutné říci, že ne každá oblast je vhodná pro nasazení této technologie a vždy je nutné pečlivě zvážit výhody a nevýhody technologie AJAX.

V obou testovaných případech rozdíly výsledků naměřených časů nebyly nijak dominantní a výrazné. Velkou výhodou však zůstává komfort a pohodlí uživatele, který s aplikací pracuje. Právě pro tyto vlastnosti technologie AJAX lze říci, že využití této technologie při tvorbě GIS aplikace je velmi vhodné a nasazením této technologie se zvýšila plynulost a efektivnost vytvořené GIS aplikace.

Seznam obrázků

Seznam tabulek

Seznam příloh

Seznam použitých zkratek

AJAX	Asynchronous JavaScript and XML
ASP	Active Server Pages
CGI	Common Gateway Interface
CSS	Cascading Style Sheets
DOM	Domument Object Model
GIS	Geographic Information Systém
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
MSIE	Microsoft Internet Explorer
PHP	Hypertext Preprocessor
PDF	Portable Document Format
URL	Uniform Resource Locator
XHTML	eXtensible HyperText Markup Language
XLink	XML Linking Language
XML	eXtensible Markup Language

Seznam použité literatury

[1] ČEPICKÝ, Jáchym. *geoinformatics.fsv.cvut.cz* [online]. 17.07.2007 [cit. 2010-04-02]. MapServer vs. Mapserver. Dostupné z WWW:<http://geoinformatics.fsv.cvut.cz/gwiki/MapServer_vs._Mapserver/>.

[2] ČEPICKÝ, Jáchym. *root.cz* [online]. 10.11.2005 [cit. 2010-04-02]. Mapový server snadno a rychle (2). Dostupné z WWW:< http://www.root.cz/clanky/mapovy-server-snadno-a-rychle-2/>.

[3] ČERVENKA, Tomáš. *Interval.cz* [online]. 09.08.2001 [cit. 2010-03-24]. Hlavičky (headers) v PHP. Dostupné z WWW: < http://interval.cz/clanky/hlavicky-headers-v-php/>.

[4] GARRETT, Jesse James. *Www.adaptivepath.com* [online]. 18.2.2005 [cit. 2010-03-24]. Ajax: A New Approach to Web Applications. Dostupné z WWW: <http://www.adaptivepath.com/ideas/essays/archives/000385.php>.

[5] JACKSON, Dean. *Www.w3.org* [online]. 05.05.2006 [cit. 2010-03-24]. The XMLHttpRequest Object. Dostupné z WWW: <http://www.w3.org/TR/2006/WD-XMLHttpRequest-20060405/>.

[6] JANOVSKÝ, Dušan. *Www.jakpsatweb.cz* [online]. 03.12.2009 [cit. 2010-03-24]. Události JavaScriptu. Dostupné z WWW: <http://www.jakpsatweb.cz/javascript/udalosti.html>.

[7] KANISOVÁ, Hana. UML srozumitelně. Brno: Computer Press, 2006. 176 s. ISBN 8025102319.

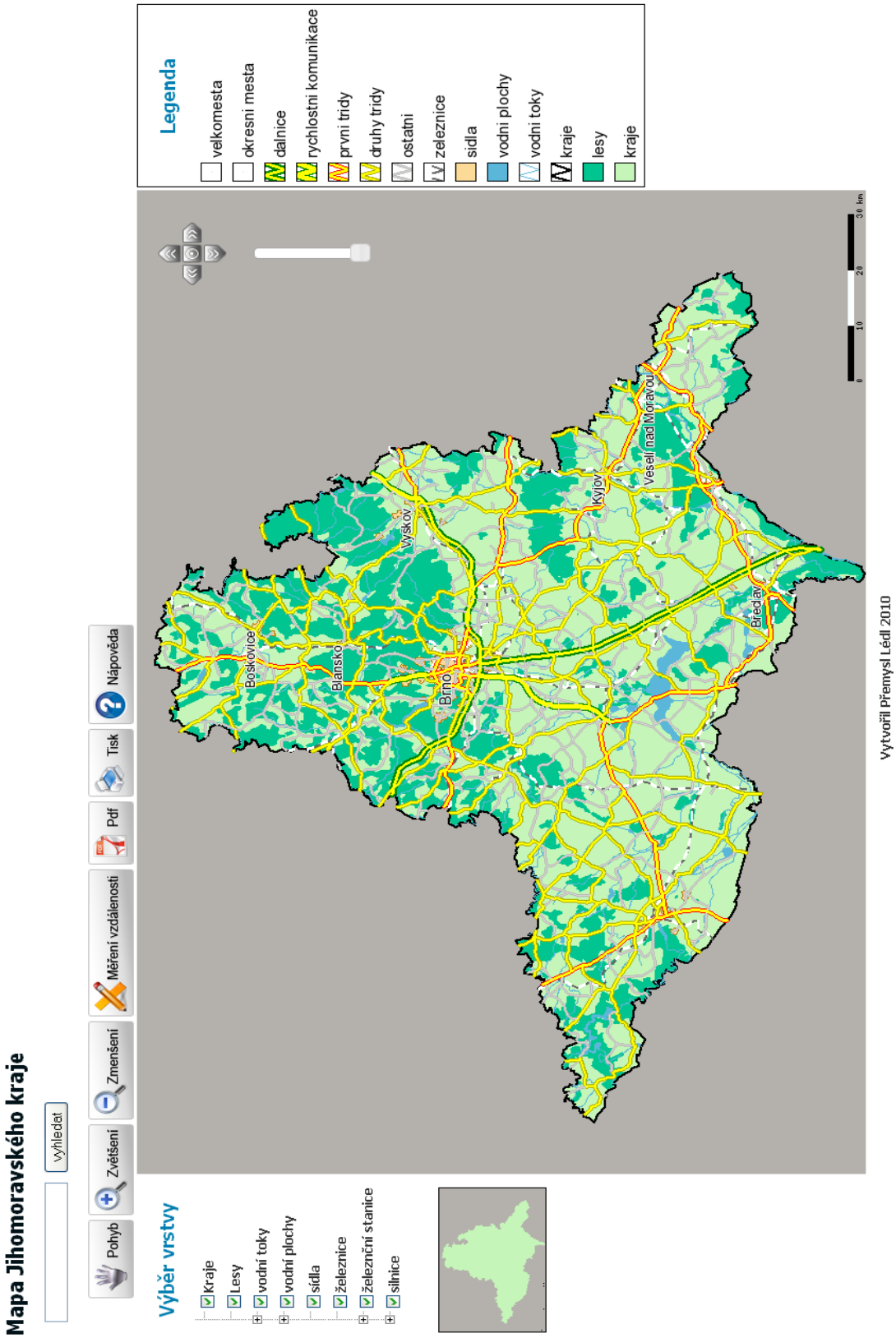
[8] KINSKÝ, Filip. *Www.dbsvet.cz* [online]. 10.2.2005 [cit. 2010-03-24]. Objektové systémy z pohledu vývojáře I. Dostupné z WWW: <http://www.dbsvet.cz/view.php?cisloclanku=2005021001>.

[9] LACKO, Luboslav. Ajax Hotová řešení. Brno: Computer Press, 2008. 272 s. ISBN 9788025121085.

[10] MUSIL, Jan. *Events.ibm-smb.cz* [online]. 22.01.2010 [cit. 2010-03-24]. Optimalizace databází. Dostupné z WWW: < http://events.ibm-smb.cz/forum2009/files/sekce2/IBM_Forum_2009_Optimalizace_Databazi.pdf>.

[11] ROBIE, Jonathan. *Www.w3.org* [online]. 27.10.2000 [cit. 2010-03-24]. What is the Document Object Model? . Dostupné z WWW: <http://www.w3.org/TR/2000/PR-DOM-Level-2-Core-0000927/introduction.html>.

Příloha 1 - Náhled webového rozhraní GIS aplikace



Příloha 2 - Zdrojové kódy webového rozhraní GIS aplikace v JavaScriptu

```
var x=Math.round((getW()-380)/2);
var y=Math.round(((getW()-380)/1.5)/2);
var extent_base="";

function send_xmlhttprequest(obsluha, method, url, content, headers)
   {
     var xmlhttp = (window.XMLHttpRequest ? new XMLHttpRequest : (window.ActiveXObject ?
new ActiveXObject("Microsoft.XMLHTTP") : false));
     if (!xmlhttp)
     { alert('chyba');
       return false;
     }
     xmlhttp.open(method, url);
     xmlhttp.onreadystatechange = function()
     {
       obsluha(xmlhttp);
     };
     if (headers) {
       for (var key in headers)
       {
         xmlhttp.setRequestHeader(key, headers[key]);
       }
     }
     xmlhttp.send(content);
     return true;
   }
function eventAjax() {
x=Math.round((getW()-380)/2);
y=Math.round(((getW()-380)/1.5)/2);
document.getElementById('mover').style.left=(getW()-480)+"px";
document.getElementById('slider-vertical').style.left=(getW()-452)+"px";
document.getElementById('image').style.height=Math.round(((getW()-380)/1.5))+"px";
selectAllLayers();

    if (!send_xmlhttprequest(reciverAjax, 'GET',
'http://cipisek.upce.cz/e078877/var1/aplikace.php?w='+(getW()-380)))
    {
      return false;
    }
    return true;
   }

function reciverAjax(xmlhttp)
   {

     if (xmlhttp.readyState == 4)
     {
            var url = xmlhttp.responseXML.getElementsByTagName('url');
            var url_ref = xmlhttp.responseXML.getElementsByTagName('url_ref');
            var url_scale = xmlhttp.responseXML.getElementsByTagName('url_scale');
```

```
            var url_legend = xmlhttp.responseXML.getElementsByTagName('url_legend');
            var ext = xmlhttp.responseXML.getElementsByTagName('extent_to_html');
            var ext_base = xmlhttp.responseXML.getElementsByTagName('extent_to_html_base');
            var x = xmlhttp.responseXML.getElementsByTagName('center_x');
            var y = xmlhttp.responseXML.getElementsByTagName('center_y');
            var st = xmlhttp.responseXML.getElementsByTagName('state');

document.getElementById('image').style.backgroundImage='url("'+url[0].firstChild.data+'")';
            document.getElementById('ref').src=url_ref[0].firstChild.data;
            document.getElementById('legend').src=url_legend[0].firstChild.data;
            document.getElementById('extent').value=ext[0].firstChild.data;
            document.getElementById('mapa_x').value=x[0].firstChild.data;
            document.getElementById('mapa_y').value=y[0].firstChild.data;
            document.getElementById('state').value=st[0].firstChild.data;
            extent_base=ext_base[0].firstChild.data;
            }
    }
function changePosionAjax(x,y,extent,zoom,keymap) {
        var kraje = (document.getElementById('kraje').checked==true?"true":'false');
        var lesy = (document.getElementById('lesy').checked==true?"true":'false');
        //vodni toky
        var toky = (document.getElementById('toky').checked==true?"true":'false');
        var toky_nazev = (document.getElementById('toky_nazev').checked==true?"true":'false');
        //vodni plochy
        var vodpl = (document.getElementById('vodpl').checked==true?"true":'false');
        var vodpl_nazev = (document.getElementById('vodpl_nazev').checked==true?"true":'false');
        var sidla = (document.getElementById('sidla').checked==true?"true":'false');
        //zeleznicni stanice
        var zel_stan = (document.getElementById('zel_stan').checked==true?"true":'false');
        var zel_stan_nazev =
(document.getElementById('zel_stan_nazev').checked==true?"true":'false');

        var zeleznice = (document.getElementById('zeleznice').checked==true?"true":'false');
        //vrstvy silnice a jeji tridy
        var silnice = (document.getElementById('silnice').checked==true?"true":'false');
        var dalnice = (document.getElementById('dalnice').checked==true?"true":'false');
        var rychlo = (document.getElementById('rychlo').checked==true?"true":'false');
        var prvni = (document.getElementById('prvni').checked==true?"true":'false');
        var druhy = (document.getElementById('druhy').checked==true?"true":'false');
        var ostatni = (document.getElementById('ostatni').checked==true?"true":'false');

        if (keymap==true) {

                var state_new =
(document.getElementById('state').value=="")?0:document.getElementById('state').value;

        }else{
                var state =
(document.getElementById('state').value=="")?0:document.getElementById('state').value;
        var state_new=parseInt(state)+parseInt(zoom);
}
if (!send_xmlhttprequest(reciverAjax, 'GET',
'http://cipisek.upce.cz/e078877/var1/aplikace.php?mapa_x='+x+'&mapa_y='+y+'&extent='+extent+'&
zoom='+zoom+'&state='+state_new+'&kraje='+kraje+'&lesy='+lesy+'&toky='+toky+'&vodpl='+vodpl
+'&toky_nazev='+toky_nazev+'&vodpl_nazev='+vodpl_nazev+'&sidla='+sidla+'&zeleznice='+zelezni
ce+'&silnice='+silnice+'&dalnice='+dalnice+'&rychlo='+rychlo+'&prvni='+prvni+'&druhy='+druhy+'
&ostatni='+ostatni+'&zel_stan='+zel_stan+'&zel_stan_nazev='+zel_stan_nazev+'&keymap='+keymap
+'&w='+(getW()-380)+"&h="+700))
```

```
   {
     return false;
       }
    return true;
  }

function getMousePosition(e,param) {
var isIE = document.all ? true : false;

var top = document.getElementById('right').offsetTop;
var left = document.getElementById('right').offsetLeft;
var extent = document.getElementById('extent').value;
            var _x;
            var _y;
            if (!isIE) {
                        _x = e.pageX-left;
                        _y = e.pageY-top;
            }
            if (isIE) {
                        _x = event.clientX-left+document.documentElement.scrollLeft;
_y = event.clientY-top+document.documentElement.scrollTop;
            }
            posX = _x;
            posY = _y;
changePosionAjax(posX,posY,extent,param,false);
}
function getMousePositionFromKeyMap(e,param) {

var isIE = document.all ? true : false;
var top = document.getElementById('ref').offsetTop;
var left = document.getElementById('ref').offsetLeft;

var extent = extent_base;
var state = document.getElementById('state').value;
var _x;
var _y;
if (!isIE) {
            _x = e.pageX-left;
            _y = e.pageY-top;
}
if (isIE) {

            _x = event.clientX-left+document.documentElement.scrollLeft;
            _y = event.clientY-top+document.documentElement.scrollTop;
}
posX = _x*Math.round((getW()-380)/150);
posY = _y*Math.round(((getW()-380)/1.5)/100);

changePosionAjax(posX,posY,extent,1,true);
            }
function getPositionForZoom(e,param) {
            var extent = document.getElementById('extent').value;
            changePosionAjax(x,y,extent,param,false);
}
function getPositionForZoomWheel(e,param) {
            var state =
(document.getElementById('state').value=="")?0:document.getElementById('state').value;
            var state_new=parseInt(state)+parseInt(param);
```

```
		if (state_new>=0 && state_new<6) {
		var isIE = document.all ? true : false;
		var top = document.getElementById('right').offsetTop;
		var left = document.getElementById('right').offsetLeft;
		var extent = document.getElementById('extent').value;
		$(function() {

			$('#slider-vertical').slider('option','value', state_new);
			$("#amount").val($("#slider-vertical").slider("value"));
		});
			var _x;
			var _y;

		if (!isIE) {
				_x = e.pageX-left;
				_y = e.pageY-top;
			}
			if (isIE) {
				_x = event.clientX-left+document.documentElement.scrollLeft;
				_y = event.clientY-top+document.documentElement.scrollTop;
			}
		posX = _x;
		posY = _y;
			changePosionAjax(posX,posY,extent,param,false);
		}
}

function getPositionForMove(e,param,number,axis) {
		var pole = new Array();
		var extent = document.getElementById('extent').value;
		var xx=0;
		switch (axis) {
		case "Z":
			xx=(x-parseInt(number));
			yy=y;
			break;
		case "V":
			xx=(x+parseInt(number));
			yy=y;
			break;
		case "S":
			yy=(y-parseInt(number));
			xx=x;
			break;
		case "J":
			yy=(y+parseInt(number));
			xx=x;
			break;
		default:
			break;
		}
		changePosionAjax(xx,yy,extent,param,false);
}

function setLayerStatus(){
		var extent = document.getElementById('extent').value;
		var kraje = (document.getElementById('kraje').checked==true?"true":'false');
```

```
		var lesy = (document.getElementById('lesy').checked==true?"true":'false');
		//vodni toky
		var toky = (document.getElementById('toky').checked==true?"true":'false');
		var toky_nazev = (document.getElementById('toky_nazev').checked==true?"true":'false');
		//vodni plochy
		var vodpl = (document.getElementById('vodpl').checked==true?"true":'false');
		var vodpl_nazev = (document.getElementById('vodpl_nazev').checked==true?"true":'false');
		var sidla = (document.getElementById('sidla').checked==true?"true":'false');
		//zeleznicni stanice
		var zel_stan = (document.getElementById('zel_stan').checked==true?"true":'false');
		var zel_stan_nazev =
(document.getElementById('zel_stan_nazev').checked==true?"true":'false');

		var zeleznice = (document.getElementById('zeleznice').checked==true?"true":'false');
		//vrstvy silnice a jeji tridy
		var silnice = (document.getElementById('silnice').checked==true?"true":'false');
		var dalnice = (document.getElementById('dalnice').checked==true?"true":'false');
		var rychlo = (document.getElementById('rychlo').checked==true?"true":'false');
		var prvni = (document.getElementById('prvni').checked==true?"true":'false');
		var druhy = (document.getElementById('druhy').checked==true?"true":'false');
		var ostatni = (document.getElementById('ostatni').checked==true?"true":'false');
		var state =
(document.getElementById('state').value==""?0:document.getElementById('state').value;
		if (!send_xmlhttprequest(reciverAjax, 'GET',
'http://cipisek.upce.cz/e078877/var1/aplikace.php?mapa_x='+x+'&mapa_y='+y+'&extent='+extent+'&
state='+state+'&kraje='+kraje+'&lesy='+lesy+'&toky='+toky+'&vodpl='+vodpl+'&toky_nazev='+toky
_nazev+'&vodpl_nazev='+vodpl_nazev+'&sidla='+sidla+'&zeleznice='+zeleznice+'&silnice='+silnice
+'&dalnice='+dalnice+'&rychlo='+rychlo+'&prvni='+prvni+'&druhy='+druhy+'&ostatni='+ostatni+'&
zel_stan='+zel_stan+'&zel_stan_nazev='+zel_stan_nazev+'&w='+(getW()-380)+"&h="+700))
   {
    return false;
   }    return true;
}
function rulers(){

		var st = document.getElementById('switcher').value;

		if (st=='0') {
			document.getElementById('switcher').value='50';
			document.getElementById('ruler_image').src='images/metr0.jpg';

		}else{
			document.getElementById('switcher').value='0';
			document.getElementById('ruler_image').src='images/metr.jpg';
			document.getElementById('zoomin_image').src='images/zoomin0.jpg';
			document.getElementById('zoomout_image').src='images/zoomout0.jpg';
			document.getElementById('move_image').src='images/move0.jpg';
		}
}
function zoomin(){

		var st = document.getElementById('switcher').value;
		if (st=='1') {
			document.getElementById('switcher').value='50';
			document.getElementById('zoomin_image').src='images/zoomin0.jpg';
		}else{
			document.getElementById('switcher').value='1';
			document.getElementById('zoomin_image').src='images/zoomin.jpg';
```

```
                document.getElementById('zoomout_image').src='images/zoomout0.jpg';
                document.getElementById('ruler_image').src='images/metr0.jpg';
                document.getElementById('move_image').src='images/move0.jpg';
        }
}
function move(){
        var st = document.getElementById('switcher').value;
        if (st=='50') {
                document.getElementById('switcher').value='20';
                document.getElementById('move_image').src='images/move0.jpg';
        }else{
                document.getElementById('switcher').value='50';
                document.getElementById('move_image').src='images/move.jpg';
                document.getElementById('zoomin_image').src='images/zoomin0.jpg';
                document.getElementById('zoomout_image').src='images/zoomout0.jpg';
                document.getElementById('ruler_image').src='images/metr0.jpg';
        }
}
function zoomout(){
        var st = document.getElementById('switcher').value;
        if (st=='2') {
                document.getElementById('switcher').value='50';
                document.getElementById('zoomout_image').src='images/zoomout0.jpg';
        }else{
                document.getElementById('switcher').value='2';
                document.getElementById('zoomout_image').src='images/zoomout.jpg';
                document.getElementById('zoomin_image').src='images/zoomin0.jpg';
                document.getElementById('ruler_image').src='images/metr0.jpg';
                document.getElementById('move_image').src='images/move0.jpg';
        }
}
function creatLinkForPDF(){
        var extent = document.getElementById('extent').value;
        var kraje = (document.getElementById('kraje').checked==true?"true":'false');
        var lesy = (document.getElementById('lesy').checked==true?"true":'false');
        //vodni toky
        var toky = (document.getElementById('toky').checked==true?"true":'false');
        var toky_nazev = (document.getElementById('toky_nazev').checked==true?"true":'false');
        //vodni plochy
        var vodpl = (document.getElementById('vodpl').checked==true?"true":'false');
        var vodpl_nazev = (document.getElementById('vodpl_nazev').checked==true?"true":'false');
        var sidla = (document.getElementById('sidla').checked==true?"true":'false');
        //zeleznicni stanice
        var zel_stan = (document.getElementById('zel_stan').checked==true?"true":'false');
        var zel_stan_nazev =
(document.getElementById('zel_stan_nazev').checked==true?"true":'false');
        var zeleznice = (document.getElementById('zeleznice').checked==true?"true":'false');
        //vrstvy silnice a jeji tridy
        var silnice = (document.getElementById('silnice').checked==true?"true":'false');
        var dalnice = (document.getElementById('dalnice').checked==true?"true":'false');
        var rychlo = (document.getElementById('rychlo').checked==true?"true":'false');
        var prvni = (document.getElementById('prvni').checked==true?"true":'false');
        var druhy = (document.getElementById('druhy').checked==true?"true":'false');
        var ostatni = (document.getElementById('ostatni').checked==true?"true":'false');

        var state =
(document.getElementById('state').value==""?0:document.getElementById('state').value;
```

```
		window.open('http://cipisek.upce.cz/e078877/var1/html_to_pdf.php?mapa_x='+x+'&mapa_y='+y+'&extent='+extent+'&state='+state+'&kraje='+kraje+'&lesy='+lesy+'&toky='+toky+'&vodpl='+vodpl+'&toky_nazev='+toky_nazev+'&vodpl_nazev='+vodpl_nazev+'&sidla='+sidla+'&zeleznice='+zeleznice+'&silnice='+silnice+'&dalnice='+dalnice+'&rychlo='+rychlo+'&prvni='+prvni+'&druhy='+druhy+'&ostatni='+ostatni+'&zel_stan='+zel_stan+'&zel_stan_nazev='+zel_stan_nazev);
}
function selectAllLayers(){
		document.getElementById('kraje').checked='true';
		document.getElementById('lesy').checked='true';
		document.getElementById('toky').checked='true';
		document.getElementById('toky_nazev').checked='true';
		document.getElementById('vodpl').checked='true';
		document.getElementById('vodpl_nazev').checked='true';
		document.getElementById('sidla').checked='true';
		document.getElementById('zel_stan').checked='true';
		document.getElementById('zel_stan_nazev').checked='true';
		document.getElementById('zeleznice').checked='true';
		document.getElementById('silnice').checked='true';
		document.getElementById('dalnice').checked='true';
		document.getElementById('rychlo').checked='true';
		document.getElementById('prvni').checked='true';
		document.getElementById('druhy').checked='true';
		document.getElementById('ostatni').checked='true';
}
function viewSearchOnMap(id,layer){

selectAllLayers();
		$(function() {
			$('#slider-vertical').slider('option','value', 4);
			$("#amount").val($("#slider-vertical").slider("value"));
		});
  if (!send_xmlhttprequest(reciverAjax, 'GET',
'http://cipisek.upce.cz/e078877/var1/aplikace.php?search_result_id='+id+"&layer="+layer+'&w='+(getW()-380)+"&h="+700))
   {
     return false;
 }
return true;
}
function searchResults(xmlhttp){
		if (xmlhttp.readyState == 4)
    {
		 var name = xmlhttp.responseXML.getElementsByTagName('name');
		 var id = xmlhttp.responseXML.getElementsByTagName('id');
		 var layer = xmlhttp.responseXML.getElementsByTagName('layer');

		changeLegendForResults(false);
   var table = "<table>";
  for ( var i = 0; i < name.length; i++) {

		 var varible=id[i].firstChild.data+",\""+layer[i].firstChild.data+"\"";
			table+="<tr onclick='viewSearchOnMap("+varible+");'>";
			table+="<td><a title='zobrazit na mapě'>"+name[i].firstChild.data+"</a></td>";
			table+="</tr>";
			}
		table+= "</table>";
		document.getElementById('table_result').innerHTML=table;
    }
```

```
}
function changeLegendForResults(value){
	if (value==true) {
		document.getElementById('legenda').style.display='block';
		document.getElementById('results').style.display='none';
	}else{
	   document.getElementById('legenda').style.display='none';
	   document.getElementById('results').style.display='block';
	}
}
function searchQueryItem(){
	var item = document.getElementById('search_text').value;
	if (item!="") {
	   if (!send_xmlhttprequest(searchResults, 'GET',
'http://cipisek.upce.cz/e078877/var1/query.php?search_item='+item))
	   {
	    return false;
	}
	return true;
	}
}
//hledani po stisknuti enter
document.onkeydown = checkKeycode;
function checkKeycode(e) {
	var keycode;
	if (window.event) keycode = window.event.keyCode;
	else if (e) keycode = e.which;
	if(keycode == 13){
		searchQueryItem();
	}
}
//rozmery prohlizece
function getW(){
	var w;
	if(document.innerWidth){ w=document.innerWidth;
	} else if(document.documentElement.clientWidth){
w=document.documentElement.clientWidth;
	} else if(document.body){ w=document.body.clientWidth; }
	return w;
}
function getH(){
	var h;
	if(document.innerHeight){ h=document.innerHeight;
	} else if(document.documentElement.clientHeight){
h=document.documentElement.clientHeight;
	} else if(document.body){ h=document.body.clientHeight; }
	return h; }
```