UNIVERZITA KOMENSKÉHO V BRATISLAVE

FAKULTA MATEMATIKY FYZIKY A INFORMATIKY

Moderné trendy pri tvorbe webových aplikácií

Bratislava 2007 Miloš Homola

Moderné trendy pri tvorbe webových aplikácií

DIPLOMOVÁ PRÁCA

Miloš Homola

UNIVERZITA KOMENSKÉHO V BRATISLAVE

FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY

KATEDRA INFORMATIKY

Odbor Informatika

Vedúci záverečnej práce

RNDr Richard Ostertág

BRATISLAVA 2007

Čestne prehlasujem, že som predloženú diplomovú prácu spracoval samostatne, pod vedením vedúceho diplomovej práce, s použitím uvedenej literatúry a ďalších informačných zdrojov.

V Bratislave 7. 5. 2007

….....

Poďakovanie

Chcel by som poďakovať všetkým, ktorí mi akýmkoľvek spôsobom pomohli pri spracovaní tejto diplomovej práce, ako aj tým vďaka ktorým som sa dostal až k tomu aby ju vôbec mohol začať písať.
Moje poďakovanie patrí môjmu diplomovému vedúcemu RNDr Richardovi Ostertágovi, za vedenie.
Osobitné poďakovanie patrí mojim rodičom, bez ich podpory a pomoci by som to určite nedokázal.

Ďakujem

Abstrakt

HOMOLA Miloš : Moderné trendy pri tvorbe webových aplikácií

Univerzita Komenského v Bratislave, Fakulta matematiky, fyziky a informatiky,
Katedra infromatiky
Školiteľ : RNDr. Richard Ostertág
Diplomová práca

Rozvoj a popularita webových aplikácií v posledných rokoch mala za následok vznik nových prístupov na ich tvorbu. Táto práca mala za cieľ zhrnúť a porovnať najčastejšie používané (AJAX, Flash) a najperspektívnejšie (XUL, XAML) technológie na ich tvorbu a porovnať ich možnosti a obmedzenia. Čitateľ, ktorý má v pláne vytvoriť webovú aplikáciu by po prečítaní tejto práce mal mať dostatok informácií aby si vedel vybrať najvhodnejšiu technológiu, v ktorej danú aplikáciu vytvorí.

Pre najrozšírenejšiu technológiu – AJAX, práca obsahuje viacero odporúčaní ako riešiť najčastejšie problémy.

Kľúčové slová: webové aplikácie, AJAX, XUL, XAML, Silverlight, Flash

Abstract

HOMOLA Miloš : Modern approach to webapplications development

Comenius University in Bratislave, Faculty of Mathematics, Physics and Informatics,
Department of Infromatics
Lector : RNDr. Richard Ostertág
MSc thesis

Expansion and popularity of web applications in recent years has brought new approaches to their development. Goal of this document is to compare most recently used development technologies (AJAX, Flash) and most perspective ones (XUL, XAML), their possibilities and limitations. Readers, who plan to develop web applications, should able to choose the most suitable technology for their development after reading this document.

For the most famous technology – AJAX, this thesis contains several recommendations to solve most common

Keywords: webapplications, AJAX, XUL, XAML, Silverlight, Flash

Obsah

Slovník pojmov a skratiek:.....9
1 Úvod.....15
1.1 História webu:.....15
1.2 Nástup webaplikácií.....15
2 Technológie na tvorbu webaplikácií.....18
2.1 AJAX (Asynchronous JavaScript nad XML).....18
2.1.1 Základná charakteristika.....18
2.1.2 Výhody a nevýhody:.....20
2.1.3 Frameworky.....23
2.1.4 SVG (Scalable Vector Graphics) - rozšírenie možností AJAXu.....23
2.2 Flash.....26
3.2.1 Základná charakteristika.....26
3.2.2 Výhody a nevýhody.....27
3.2.3 Frameworky.....29
2.3 XAML (eXtensible Application Markup Language).....30
2.3.1 Základná charakteristika.....30
2.3.2 XBAP (XAML Browser Application).....32
2.3.3 Silverlight - WPF/E (Windows Presentation Foundation for Everyvhere).....33
2.3.4 Prípadová štúdia možností XAML.....35
2.3.5 XAML a obvinenie Microsoftu.....36
2.3.6 Výhody a nevýhody.....37
2.3.7 Bezpečnosť.....38
2.4 XUL (XML-based User-interface Language).....39
2.4.1 Základná charakteristika.....39
2.4.2 Výhody a nevýhody.....42
2.5 Java Applety.....42
2.5.1 Základná charakteristika.....42
2.5.2 Výhody a nevýhody.....44
2.6 Štandardy vo vývoji.....45
2.6.1 Web Applications 1.0 (HTML5).....45
3 Porovnanie spomenutých technológií.....46
3.1 Bezpečnosť.....46
3.2 Rýchlosť vývoja.....46

3.3 Platformová nezávislosť 47
3.3.1 Z pohľadu operačného systému 47
3.3.2 Z pohľadu prehliadača 48
3.4 Znevýhodnené skupiny užívateľov 50
3.5 Hranice jednotlivých technológií a ich najvhodnejšie uplatnenie 51
3.5.1 AJAX 51
3.5.2 FLASH 52
3.5.3 XAML, Silverlight 53
3.5.4 XUL 53
3.5.5 Java applety 53
4 Aktuálny stav používania technológií 54
4.1 Technológie používané pri tvorbe internetových stránok 54
5.2 Prieskum medzi 5000 americkými vývojármi, aké technológie sa využívajú v ich firmách pri tvorbe internetových projektov 55
5.3 A aké plánujú využívať v budúcnosti 56
5 Ako na najčastejšie problémy AJAXu 57
5.1 Uchovanie stavu 57
5.1.1 Uloženie stavu na strane klienta 59
5.1.1.1 Uloženie stavu v URL 59
5.1.1.2 Uloženie stavu v cookies. 60
5.1.1.3 V obsahu stránky 60
5.1.1.4 V súborovom systéme užívateľovho počítača 60
5.1.1.5 Vo Flashi 60
5.1.2 Na strane servera 61
5.2 Ošetrenie stlačenia tlačidla „späť“ 61
5.3 Pridanie medzi obľúbené a posielanie linkov 62
5.4 Ak sa niečo vykonáva, alebo čaká na server užívateľ by to mal vedieť 63
5.5 Bezpečnosť a dôvernosť 63
5.6 Neočakávané správanie a funkcie sa užívateľského rozhrania 64
6 Záver 65
6.1 Microsoft 65
6.2 Google 66
7 Pohľad na budúcnosť (web)aplikácií 68
8 Použitá literatúra 71

Slovník pojmov a skratiek:

.NET – termín zastrešujúci široké spektrum produktov od Microsoftu

action script - programovací jazyk používaný vo Flashi

API - (Application Programming Interface) množina funkcií zabezpečovaná systémom, alebo programovacím jazykom

Architektúra klient - server . Systémy založené na tejto architektúre rozdeľujú prácu jednej aplikácie medzi dva programy - klienta a server, ktoré spolu vedú dialóg prostredníctvom siete. Program klienta zistí podrobnosti o tom, čo potrebujeme (informácie, prístup k určitému zdroju a pod.) a vytvorí spojenie s programom servera, ktorý spravuje hľadané informácie. Pod pojmom server chápeme tak program, ako aj samotný počítač, ktorý poskytuje disk, súbory, tlačiarne a pod. ostatným počítačom siete. Klientské programy bežia náhodile. Spúšťajú sa vždy, keď klient potrebuje nejakú informáciu (odpoveď na otázku). Klientský program sa zastaví, keď dostaneme požadovanú informáciu, alebo zdroj. Servery bežia nepretržite, čakajú, kedy sa na nich klienti so svojimi požiadavkami obrátia. Programy klient - server bežia na jednom počítači i na vzdialených počítačoch.

Bohaté užívateľské rozhranie – (Rich user interface - RIA) pokročilé užívateľské rozhranie obsahujúce resp podporujúce množstvo grafických elementov napríklad záložky, kontextové menu, umožňujúce presúvať prvky (drag and drop), atď

baner – grafická plocha, zväčša animovaná, slúžiaca na spestrenie vzhľadu, no najčastejšie ako reklama

base64 – spôsob konverzie binárnych dát, do textovej formy

BPEL - (Business Process Execution Language) jazyk na popis správania sa aplikácie

bytecode – binárna reprezentácia spustiteľného programu spustiteľná v virtuálnom stroji

(napr. JVM)

CERN – Európsky ústav pre nukleárny výskum, položili v ňom základy internetu tak ako ho poznáme

CLR - (Common Language Runtime) virtuálny stroj, ktorý dokáže pracovať so všetkými jazykmi patriacimi pod .NET, v takto naprogramovaných aplikáciách je následne možné využívať služby a volať funkcie aj v iných programovacích jazykoch naprogramovaných komponentoch tejto aplikácie

cookies – umožňuje serveru poslať klientovi dáta, ktoré si ten uloží, buď dočasne, lebo navždy a zakaždým keď odosiela požiadavku na server tak ich spolu s touto požiadavkou pošle

CSS – (Cascading Style Sheets) rozšírenie HTML umožňujúce oštýlovať jednotlivé elementy

desktopová aplikácia – aplikácia bežiaca celá na počítači užívateľa, opak v porovnaní s aplikáciou klient-server

DOM – (Document Object Model) na platforme a programovacom jazyku nezávislý štandard, na reprezentáciu XML dokumentov

DLR - (Dynamic Language Runtime) táto technológia umožňuje dynamickým jazykom sa začleniť do CLR

Flash - technológia na tvorbu webaplikácií, podrobnejšie popísaná v kapitole 3.2

Framework- viacnásobne použiteľný prístup, alebo program pri vývoji softvéru, jeho použitie zvyčajne zjednodušuje prácu

FTTH – (Fibre To The Home) technológia na pripojenie k internetu pomocou optických káblov, umožňuje najvyššie možné rýchlosti

Gecko - *(Gecko Runtime Enviroment)* knižnica na popis grafického rozhrania s množstvom API využiteľným predovšetkým pri internetových aplikáciách

hypertext – text na ktorý sa dá kliknúť a zobrazí sa ďalšia informácia, zväčša relevantná k tomuto textu

HTML- (HyperText Markup Language) dominantný jazyk na popis štruktúry internetových stránok

HTTP – (HyperText Transfer Protocol) protokol na komunikáciu medzi klientom a serverom používaný na internete

implementácia – naprogramovanie aplikácie, alebo jej časti

interaktívne aplikácie – aplikácie ktoré užívateľovi dávajú okamžitú spätnú väzbu

Internet - globálna sieť umožňujúca komunikáciu cez viacúrovňové počítačové siete.

Java Applet – nástroj umožňujúci spúšťať v internetovom prehliadači v Jave napísané aplikácie pomocou JVM

JVM – (Java Virtual Machine) počítačový program, v ktorom sú spúšťané v Jave napísané aplikácie

JavaScript – najpoužívanejší programovací jazyk používaný na internetových stránkach, na strane klienta

JSON – textový formát na prenos dát, na rzodiel od XML neobsahuje informáciu o prenášaných dátach, ale len samotnú informáciu

klient - počítač, ktorý pristupuje k zdieľaným sieťovým zdrojom, ktoré ponúka iný počítač (server).

kodek – program na prehrávanie digitálnych dát, zväčša videa

odozva systému - časový interval vymedzený medzi aktivitou užívateľa pri zadávaní požiadavky do obdržania odpovede.

PDA – (Personal Digital Assistant) malé a ľahké zariadenie s mnohými funkciami veľkého počítača

platforma – typ hardvéru, alebo softvéru na ktorom je možné spúšťať aplikácie

PHP – programovací jazyk používaný na programovanie na strane servera

rastrová grafika – spôsob uchovania obrazových dát ako mapu bodov určitej farby, opak vektorovej grafiky

RuleML – (Rule Markup Language) jazyk na popis správania aplikácií

serializovaný objekt – objekt, hodnota jeho vlastností a celý jeho stav vyjadrený v textovej, alebo binárnej forme, v takej to forme sa dá objekt uložiť a neskôr obnoviť

server - počítač, ktorý poskytuje sieťovým používateľom zdieľané zdroje.

session – pole premenných uchovávaných na serveri, po vypršaní časového limitu dôjde k ich vymazaniu, v PHP je tento časový limit štandardne 3 hodiny

SOA - (Service-Oriented Architecture) architektonický model pri ktorom sú jednotlivé služby zabezpečované zväčša cez sieť, programátora klientskej aplikácie nezaujíma, ako je daná služba naprogramovaná, len jej pošle požiadavku a požaduje odpoveď

tag – reťazec používaný v predovšetkým v HTML a XML na reprezentáciu začiatku a konca elementu

tenký klient – program, alebo hardvérové zariadenie, ktorý väčšinu funkcií prenecháva na server a samo má na starosti len zobrazovanie výstupu užívateľovi a posielanie užívateľom zadaných požiadaviek na server, nespracováva žiadne alebo len minimum informácií

trojvrstvová architektúra – architektonický model používaný predvšetkým pri vývoji klient-server aplikácií, oddeľuje prezentačnú, logickú a dátovú zložku

URI – (Uniform Resource Identifier) všeobecná množina všetkých mien a adries určujúcich na nejaké objekty

URL – (Uniform Resource Locator) štandardný spôsob ako určiť umiestnenie objektu, zväčša internetovej stránky na internete, podmnožina URI

W3C - (World Wide Web Consortium) hlavná organizácia zaoberajúca sa štandardizáciou protokolov na internete

Webový štandard – súbor noriem schvaľovaných W3C konzorciom, všetky aplikácie pracujúce s internetom by ich mali dodržiavať, najstaršie a najpodstatnejšie sú URI, HTTP, HTML

webservice – softvérový systém dizajnovaný na podporu spolupráce rôznych zariadení cez sieť

WF – (Windows Workflow Foundation) technológia Microsoftu na definovanie, vykonávanie a riadenie. Je súčasťou .NET Frameworku 3.0

WHATWG - (Web Hypertext Application Technology Working Group) pracovná skupina spolupracujúca s W3C na vývoji nových technológií určených pre web umožňujúcich programátorom vyvíjať webaplikácie jednoduchšie, členmi sú o.i. Google, Mozilla Foundation, Microsoft, Opera Software a Apple Inc

WPF – (Windows Presentation Foundation) názov grafického subsystému .NET Frameworku 3.0, nazývaného tiež Avalon

WPF/E – (Windows Presentation Foundation for Everywhere) mlutiplatormová podmnožina WPF, nazývaná tiež Silverlight

WWW - World Wide Web - multimediálny systém umožňujúci vyhľadávanie a prezeranie informácií na Internete.

XBAP - (XAML Browser Applications) na XAML postavená technológia na tvorbu webaplikácií, podrobnejšie v kapitole 3.3.2

XML – (eXtensible Markup Language) jazyk na popis dát, využívaný predovšetkým na ich výmenu medzi rôznymi aplikáciami

XMLHttpRequest – API funkcia JavaScriptu umožňujúca mu komunikovať so serverom bez nutnosti načítania celej stránky

XSLT – (eXtensible Stylesheet Language Transformations) jazyk na zmenu jedného XML dokumentu na iný, aj XSLT je postavené na XML

XUL - (XML-based User-interface Language) jazyk na popis grafického prostredia, podrobnejšie popísanýv kapitole 3.4

1 Úvod

1.1 História webu:

Koncom roka 1990 Tim-Berners Lee vytvoril v CERNe 1. web server, webový prehliadač (obsahoval aj editor) a 1. web stránku, ktorá bola venovaná tomuto projektu.
6. augusta 1991 T.B.Lee poslal do diskusného fóra alt.hypertext stručný popis projektu WorldWideWeb (www). Tento dátum sa dá považovať za začiatok internetu tak ako ho poznáme dnes. Do vtedy to bola zmes rôznorodých štandardov a spoločné rozhranie na zdieľanie hypertextových informácií neexistovalo.

Internet sa časom stal prostriedkom nielen na získavanie informácií na čo bol primárne určený a vytvorený, ale aj obyčajný ľudia ho začali využívať na ich zdieľanie a prácu s nimi. HTML, ako základný štandard na prezentáciu dát na internete, bolo pri svojom návrhu vytvorený ako štandard na prezentáciu dát a nie na interaktívnu prácu s nimi. Aby sa v ňom dalo vytvoriť interaktívne a užívateľsky príjemné užívateľské rozhranie je potrebné použiť niečo viac než len HTML.

Možností z ktorých sa dá vyberať je viacero. Klasické HTML už nestačí, niekoľko sekundová odozva servera a čakanie na zobrazenie stránky je príliš veľa a užívateľ by najradšej pracoval tak ako je zvyknutý z desktopovej aplikácie.

1.2 Nástup webaplikácií

Webaplikácie sú aplikácie spúšťané v internetovom prehliadači zo siete. Prehliadač plní pri takomto type aplikácií funkciu tenkého klienta, ktorý však väčšinou nieje úplne tenký a vykonáva časť programovej logiky sám. Svoju popularitu si získavajú predovšetkým tým že internetový prehliadač je dnes dostupný v podstate kdekoľvek. Tak na prácu s

webaplikáciou stačí už len pripojenie na internet. Nieje nutná žiadna inštalácia stačí si len otvoriť správnu internetovú stránku. Zbavenie sa nutnosti inštalovať programy na potenciálne stovky alebo tisícky počítačov ocenia predovšetkým vo veľkých firmách, v ktorých nasadenie nového softvér na tak veľké množstvo počítačov vyžaduje desiatky hodín práce administrátorov. Aj aktualizácia a pridávanie novej funkcionality je v porovnaní s desktopovými aplikáciami jednoduchšie, stačí spraviť zmenu len raz na serveri a všetkým používateľom sa okamžite prejaví.

Predchádzajúce typy klient-server aplikácií mali každá svoju vlastnú klientskú časť. A spravidla bolo nutné v prípade aktualizácie aplikácie na serveri aktualizovať aj programy na počítačoch užívateľov, ktoré s nim komunikovali. V porovnaní s tým webaplikácie vytvárajú obsah v štandardizovanom formáte, ktorý dokážu zobrazovať zvyčajne všetky internetové prehliadače.

Už pár rokov môžeme sledovať postupný nástup webových aplikácií, ktoré majú úspech u užívateľov najmä preto že sa ich užívateľské rozhranie interaktivitou blíži k desktopovým aplikáciám. Užívatelia takisto oceňujú že nie sú viazaní na počítač ktorý majú doma, v práci, alebo kdekoľvek inde. Na používanie im stačí počítač s pripojením na internet a internetový prehliadač, poprípade ďalšie programové vybavenie nutné pre technológie použité pri tvorbe tejto aplikácie, napríklad .NET Framework pre aplikácie založené na XAML alebo Flasha pod.

Z programátorského pohľadu sa stretávame sa so snahou urobiť v momentálnych technologických odmienkach čo možno najpohodlnejšie použiteľné aplikácie, ktoré by boli dostupné vždy a všade. Aktuálnym lídrom v tom segmente je Google, ktorý sa z vyhľadávača prepracoval na firmu ktorá vytvára užívateľsky jednoduché no komplexné a interaktívne webové aplikácie, založené na štandardoch ktoré sú otvorené a implementované na všetkých bežne používaných platformách.

Na stránke projektu eyeos (5) je možné nájsť víziu toho kam to s webaplikáciami

môže dospieť. Kompletný operačný systém v okne internetového prehliadača obsahujúci nástroj na písanie dokumentov s funkciami ako poznáme z Wordu, ICQ klient, prehrávač videa, kalkulačka, nástroj na kreslenie, samozrejme aj hry a množstvo ďalších programov.

Výhodou webaplikácií oproti klasickým internetovým stránkam je že nieje nutné znovu načítať celú stránku pri každej udalosti ktorú návštevník na stránke vykoná. Napríklad keď sa na stránke nachádza kalendár a užívateľ si chce pozrieť nasledujúci mesiac, tak sa zo servera načítajú len dáta, ktoré Javascript potrebuje, aby vedel správne tento mesiac zobraziť a užívateľ nepríde o rozpísaný text, pozíciu na stránke na ktorej je posunutý a podobne.

Užívateľ tým získa pocit väčšieho pohodlia, keďže takáto stránka sa viac podobá desktopovým aplikáciám a ich chovaniu, na ktoré je zvyknutý.

Výhody sú aj na strane servera, keďže stránka sa nemusí zakaždým generovať celá, jej ďalšie vygenerovanie je jednoduchšie a rýchlejšie a server dokáže obslúžiť väčšie množstvo užívateľov. Nie je to však pravidlo, ak je takáto aplikácia zle navrhnutá môže generovať príliš veľa požiadaviek na server a jeho záťaž by sa istých okolností mohla byť aj vyššia ako pri klasických, statických stránkach.

2 Technológie na tvorbu webaplikácií

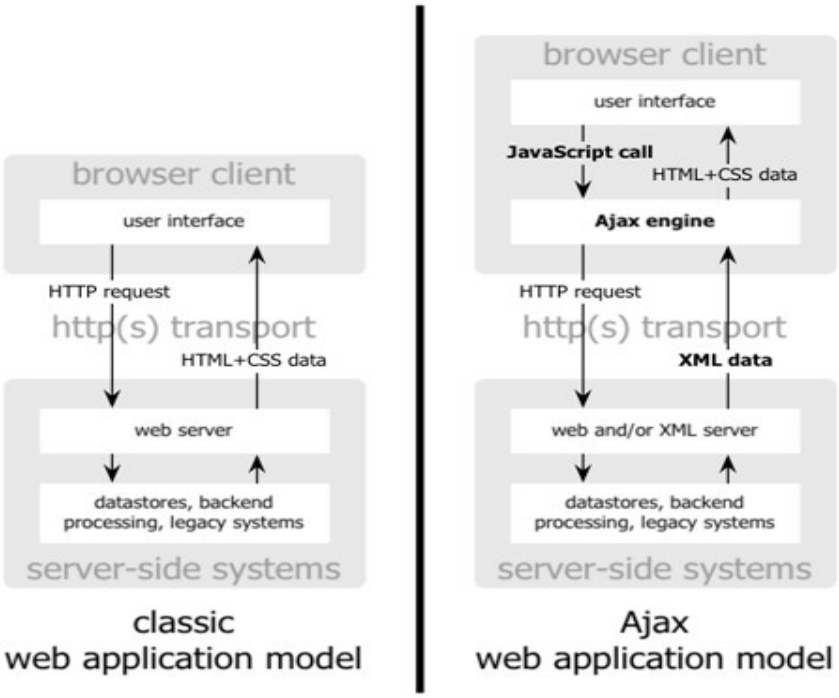
2.1 AJAX (Asynchronous JavaScript nad XML)

2.1.1 Základná charakteristika

Momentálne je AJAX najrozšírenejšia technológia na tvorbu webových aplikácií. Nejedná sa o technológiu v klasickom zmysle tohto slova, ide skôr o názov pre skupinu techník spoločne využívaných na vytvorenie užívateľsky prívetivej webovej aplikácie.

Názov vymyslel Jesse James Garrett vo Februári 2005 v článku (4) o novom prístupe k webovým aplikáciám niekoľko rokov potom ako sa objavili prvé aplikácie ktoré tieto technológie takýmto spôsobom používali. Dovtedy tento prístup nemal spoločný názov a v tom čase bolo veľmi málo firiem, ktoré by vo svojich aplikáciách využívali JavaScript ako plnohodnotný programovací jazyk. Dva roky od vtedy má vyhľadávač Google pri slove AJAX zaindexovaných takmer 70 miliónov stránok.

Princíp je jednoduchý stránka sa skladá z normálneho (x)HTML a JavaScriptu. A aby nebolo pri každom kliku potrebné načítavať celú stránku a čakať na zobrazenie novej stránky niekoľko sekúnd, JavaScript odchytí kliknutie, ak má dostatok informácií tak dá užívateľovi okamžite spätnú väzbu, ak informácií dostatok nemá, môže poslať požiadavku na server, spracovať odpoveď a na jej základe zmeniť vzhľad stránky. (Obr. 1: AJAX - porovnanie s klasickými stránkami)



Obr. 1: AJAX - porovnanie s klasickými stránkami

Ako som už spomenul AJAX nieje technológia v obvyklom zmysle toho slova, je to spojenie viacerých technológií do jedného celku, ktorý sa správa nejakým spôsobom. Do tohto procesu vstupujú nasledovné technológie ktoré pri vhodnom spojení dokážu dať užívateľovi novú skúsenosť s používaním internetových stránok:

(x)HTML – prezentované dáta ktoré sa v prehliadači zobrazia

CSS – popis vzhľadu

DOM – medzistupeň pri aplikácii zmien medzi JavaScriptom a (x)HTML

XML – formát komunikačných dát

XSLT – transformácie XML na HTML

XMLHttpRequest – spôsob komunikácie so serverom

JavaScript – jadro celej interaktivity

Od počiatku bola skratka AJAX kritizovaná za X na svojom konci, pretože formát

komunikačných dát nieje vôbec obmedzený na XML a medzi prehliadačom a serverom sa informácie dajú prenášať niekoľkými spôsobmi. Výhodou ak formátom na prenos dát je XML je, že takéto dáta sa dajú ľahko čítať, pretože formát XML okrem samotných informácií obsahuje aj popis čo tieto informácie znamenajú. To však okrem programátora tejto, alebo túto službu využívajúcej web aplikácie nikto iný nevyužije. Ak by sa ako formát komunikačných dát použili napríklad JSON, v závislosti od konkrétneho prípadu možno ušetriť aj značne veľké množstvo dát, ktoré sú v XML použité na reprezentáciu XML tagov. V prípade JSONu nieje prenášaná aj informácia o informácii, ako v XML, ale len informácia samotná. Takisto je možné zo servera poslať priamo JavaScriptovú funkciu, ktorá sa v prehliadači vykoná, alebo HTML kód ktorý sa dá priamo vložiť na konkrétne miesto v stránke.

1. väčšie využitie tejto technológie prišlo v roku 2000, keď Microsoft využil XMLHttpRequest vo svojom produkte Outlook Web Access, ktorý poskytoval web rozhranie na prístup k emailom na Microsoft Exchange Serveri cez Internet Explorer 5.(2)

Väčšiu popularitu však získaval až po roku 2004, významný pokrok v jeho spopularizovaní mala niekoľkými svojimi službami spoločnosť Google: Gmail, Google Maps. Ktorá naďalej rozširuje svoje produkty o ďalšie webaplikácie.

Pri písaní tejto práce ma napadlo že je škoda, že riešenie typu AJAX prišlo až v čase keď rýchlosť a ani množstvo prenesených dát už nie sú taký problém ako kedysi. V časoch keď internet pred 15 rokmi začínal by sa veľmi zošlo keby sa nemuseli načítavať celé stránky, ale JavaScriptom len potrebné časti.

2.1.2 Výhody a nevýhody:

V porovnaní s ostatnými technológiami spomenutými v tomto dokumente má práve AJAX najširšiu podporu medzi prehliadačmi a až na zopár výnimiek (mobilné telefóny s

podporou Flash), ak prehliadač nepodporuje AJAX nebude podporovať ani ostatné technológie.

AJAX ako jediný je plne kompatibilný s webovými štandardami, keďže je z nich odvodený.

Existuje množstvo frameworkov ktoré uľahčujú vývoj takýchto aplikácií a riešia aj mnohé problémy, ktoré sú s AJAXom spojené.

Kompletný zoznam nevýhod, alebo skôr problémov spojených s AJAXom je zhrnutý v blogu Alexa Boswortha (3).

Pomerne veľká skupina nevýhod plynie z výhod ktoré AJAX ponúka. Keď JavaScript zmení obsah stránky a užívateľ sa chce vrátiť o krok späť intuitívne použije tlačidlo „Späť“, tak ako bol z internetových stránok doteraz zvyknutý, prehliadač ho síce vráti späť o jeden krok, ten však z pohľadu užívateľa môže byť krokom späť aj o niekoľko desiatok krokov a dobrý dojem z takejto stránky je zrazu preč, ak sa užívateľovi nepodarí dostať do toho stavu v ktorom pre stlačením toho tlačidla bol. Podobne si užívatelia zvykli kopírovať a posielať adresy stránok, alebo si ich ukladať do obľúbených. Tento problém je z programátorského hľadiska riešiteľný ale treba naň myslieť a naprogramovať ho v opačnom prípade bude užívateľ možno chvíľu nadšený z toho ako to pekne funguje, ale možno by to radšej oželel, keby si mohol vytvoriť záložku. Na druhej strane pri desktopových aplikáciách také tlačidlo ako „späť“, nemyslím tým „undo“, nieje.

Problémom sú aj užívatelia ktorí nie sú pripojení na internet, alebo tí ktorí si takúto stránku pokúšajú otvoriť z viac či menej exotických zariadení a prehliadačov. Vývojári internetových stránok veľmi dobre poznajú problémy spojené s tým aby stránka vyzerala rovnako aspoň v najpoužívanejších prehliadačoch. Rôzne chápanie jazyka HTML prehliadačmi pritom nie je najväčším problémom. Pri AJAXe sa vyskytujú ďalšie problémy pri rôznych implementáciách DOMu a JavaScriptu. A bolo by vhodné nezabúdať aj na užívateľov ktorí používajú prehliadač bez podpory JavaScriptu v prípade že je táto

technológia použitá na normálnej internetovej stránke.

Ak by sa jednalo o webaplikáciu pre konkrétnu firmu a jej zamestnancov tak by sa tieto problémy dali vyriešiť určením platformy, na ktorej bude táto web aplikácia plne funkčná a ostatné programátorov nemusia zaujímať. Avšak AJAX nachádza svoje miesto veľmi často na obyčajných stránkach. Vtedy je potrebné podrobné testovanie aby nebolo viac nespokojných užívateľov, ako tých ktorým sa stránka bude zobrazovať správne. Obvykle sa stránky optimalizujú pre Internet Explorer a Firefox, čo by sa pri statických HTML stránkach mohlo považovať možno aj dostatočné. Vo všetkých ostatných, i keď stránka nebude vyzerať na pixel presne ako dizajnový návrh, tak bude čitateľná a použiteľná. Toto však pri AJAXových stránkach neplatí. Ak JavaScript nefunguje správne tak to môže znamenať že nefunguje stránka ako celok.

Ďalším problémom je že užívatelia si už zvykli na určité správanie sa prehliadačov a stránok v nich zobrazovaných. Dynamická zmena obsahu, v časti stránky ktorú už užívateľ nemusí mať viditeľnú na onraozvke, preto takisto nieje dobrý spôsob ako využiť možnosti ktoré AJAX ponúka.

Rastie tiež počet ľudí ktorí surfujú po internete z mobilných telefónov či PDA zariadení, v ktorých (zatiaľ) nie sú plnohodnotné prehliadače a ak tam aj v budúcnosti budú veľkosť zobrazovacieho zariadenia ostane malá a treba s tým počítať.

Nielen kvôli užívateľom s neštandardným prehliadačom ale aj kvôli vyhľadávačom by sa nemalo zabúdať na verziu stránky ktorá bude fungovať aj bez JavaScriptu.

Nevýhodou ktorá sa pri aplikácii ktorá má snahu blížiť sa svojim správaním desktopovej je podpora multimédií a práca s grafikou, ktorá je limitovaná možnosťami HTML.

2.1.3 Frameworky.....

Keďže vývoj AJAXových aplikácií je pomerne náročný na testovanie kompatibility medzi jednotlivými prehliadačmi, vznikli desiatky(55) frameworkov a knižníc, ktoré sa prácu s ním snažia uľahčiť. Mnohé z nich fungujú na princípe, že programátor vytvorí užívateľské prostredie v pre daný framework zodpovedajúcom XML formáte, ktorý je ešte na serveri spracovaný a do užívateľovho prehliadača putuje už na jeho základe vygenerované HTML a JavaScript, lebo len JavaScript.

Spočiatku sa mi nápad poslať zo servera len JavaScript a až v prehliadači z neho vygenerovať HTML páčil. Keďže Javascript celú stránku vygeneruje má nad ňou úplnú kontrolu a funkcie na prácu a vykreslenie každého elementu. Nieje nutné dávať si pozor či pri zmene HTML je potrebné zodpovedajúcim spôsobom zmeniť aj JavaScript. No tento spôsob úplne znemožňuje používať takúto aplikáciu užívateľom bez JavaScriptu, alebo v nepodporovanom prehliadači.

Okrem toho existujú nástroje ako Google Web Toolkit, alebo od Microsoftu AJAX.NET (pôvodne projekt Atlas), ktoré umožňujú program napísaný v Jave, resp. .NETe prekonvertovať na AJAXovú aplikáciu aj s k tomu prislúchajúcou serverovou časťou.

2.1.4 SVG (Scalable Vector Graphics) - rozšírenie možností AJAXu

Možnosti AJAXu sú pri práci s grafickými elementami značne obmedzené. Dokáže presúvať jednotlivé elementy a meniť im niektoré základné vlastnosti ako veľkosť, polohu, alebo ich zo stránky odstrániť, no na prácu s grafikou nemá žiadne nástroje.

Možnosťou ako dostať z prehliadača a AJAXu viac než sú jeho štandardné možnosti je SVG. Nieje to síce priamo nástroj na tvorbu web aplikácií. No využitie vo webaplikáciách v spolupráci s AJAXom alebo ActiveX si nájde. Hlavným uplatnením tejto

technológie je vektorová grafika a prezentácia numerických dát v grafickej podobe interaktívnym spôsobom na internetových stránkach. Svoje uplatnenie našiel aj ako jazyk na popis grafického rozhrania v niektorých nových mobilných telefónoch. (41)(45)

SVG je štandard vyvíjaný W3C predstavený v roku 1999 a verzia 1.0 prišla v roku 2001 a o 2 roky neskôr verzia 1.1, ktorá sa však od pôvodnej príliš neodlišuje, hlavný rozdiel je že umožňuje definovať podmnožinu tohto štandardu. Na základe tohto rozšírenia sa v roku 2003 sa objavili aj verzie SVG Tiny a SVG Basic. Tieto podmnožiny vznikli na požiadavku trhu mobilných zariadení, ktoré majú obmedzené zobrazovacie schopnosti. Verzia Tiny je pre mobilné telefóny a Basic pre mobilné zariadenia vyššej kategórie, ako sú napríklad vreckové počítače (PDA).(44)

SVG sa zameriava sa na dvojrozmerné zobrazovanie a je jednoducho vložitelný do HTML stránky. Tým čo ponúka je výborným doplnkom k textovým informáciám ktoré sa najlepšie reprezentujú práve pomocou HTML. Interakcia s užívateľom je zabezpečená pomocou JavaScriptu. Tak ako pri HTML aj SVG fragmenty môžu na seba odkazovať pomocou tagu „a href“. (39)

Aby používateľ Internet Explorera bol schopný vidieť v SVG vytvorené grafické prvky musí si doinštalovať rozšírenie ktoré mu to umožní, v budúcnosti by mala byť podpora tohto štandardu zahrnutá aj priamo do Internet Explorera, no do verzie 7 nebola implementovaná. V Opere i Firefoxe je podpora pre tento štandard natívna no nie kompletná. Opera od verzie 8 podporuje štandard SVG Tiny 1.1. a od verzie 9 SVG 1.1 Basic. Firefox vo svojej aktuálnej verzii 2 podporuje istú podmnožinu štandardu SVG 1.1, no nieje to ani SVG Tiny 1.1 ani SVG Basic 1.1.(40) Vo verzii 3 bude podpora tohto štandardu opäť rozšírená.

V SVG vytvorené obrázky sú výrazne menšie ako klasické pre web určené formáty .jpg, .png a .gif. Za cenu toho že chýba rastrová informácia o povrchu predmetov. Takéto

obrázky je možné zväčšovať bez toho aby bola viditeľná strata kvality obrazu. Obraz je každým zobrazovacím zariadením (monitor, tlačiareň) zobrazený tak dobre ako to dané zariadene umožňuje a nieje nutné na obraz aplikovať žiadne efekty ako zväčšovanie pixelov (pixel enlargement) ani zaobľovanie hrán (aniti-aliasing). (42)

Tým že SVG je založené na XML a teda je to textový súbor je možné v ňom vyhľadávať a indexovať ho podľa informácie ktorú skutočne zobrazuje. Ako aj sprístupniť ho užívateľom s poškodením zraku.

Ďalšími dobrými vlastnosťami ktoré SVG má je že obrázok sa vie animovať – meniť, priesvitnosť je samozrejmosťou, text sa dá kopírovať (táto funkcionalita závisí od prehrávača). Exportovať do tohto formátu vie každý pokročilejší program na prácu s vektorovou grafikou.

Obr. 2: Príklad SVG

```
<rect id=”Rectangular_shape”  width=”85.302” height=”44.092”/>
<ellipse id=”Elliptical_shape”  cx=”42.651” cy=”22.046” rx=”35.447” ry=”16.426”/>
<text transform=”matrix(1 0 0 1 16.2104 32.2134)” font-family=”ÅfMyriad-RomanÅf”
font-size=”31.2342”>SVG</text>
```

Najväčšie uplatnenie má SVG pri zobrazovaní dynamicky sa meniacich dát. Keďže formát je textový nieje problém akýmkoľvek programovacím jazykom vygenerovať textový súbor ktorý sa v prehliadači interpretuje ako obrázok. V samotnom AJAXe sa s obsahom

rastrových obrázkov pracovať nedá a spolu s Flashom je SVG jediná možnosť pre AJAXové aplikácie ako pracovať s grafikou na strane klienta.

Animácie na internete si každý spája s Flashom no SVG si je v mnohých ohľadoch podobné práve s Flashom. Oba sú vektorovo orientované, môžu byť animované, môžu obsahovať zvuk (zvuk nieje priamo súčasťou špecifikácie SVG), sú skriptovatelné, SVG sa dá skriptovať pomocou JavaScriptu, Java Bindings a ActiveX. V porovnaní s Flashom má SVG najväčšiu výhodu v tom že je to XML dokument, dá sa teda jednoducho editovať a dokáže jednoducho spolupracovať s ostatnými na webových štandardoch založenými technológiami. (43)

Problémom je že podpora SVG v prehliadačoch nieje (IE bez nainštalovaného rozšírenia), alebo je úroveň kompatibility so štandardom dosť rôzna v jednotlivých prehliadačoch. Čo značne zhoršuje možnosti využitia tejto zaujímavej technológie pri tvorbe dizajnu. Podľa štatistiky (7) z marca 2007 malo prehliadač s podporou pre SVG len 10% používateľov internetu.

2.2 Flash

3.2.1 Základná charakteristika

Interaktivitu v tomto prípade namiesto JavaScriptu zabezpečuje Flash (celým menom Adobe Flash, po tom ako v roku 2005 Adobe kúpil Macromediu sa Macromedia Flash premenoval). Flash v porovnaní s AJAXom nepracuje s HTML stránky na ktorej sa nachádza, je to do stránky vložený prvok ktorý mení a interakciu poskytuje v sebe samotnom.

Macromedia ako tvorca Flashu už vo verzii 4 prišla s podporou posielania

požiadaviek na server a ich následného spracovávania vo Flashi bez nutnosti znovu načítania celej stránky. Dnes už existuje verzia 9 so skriptovacím jazykom ActionScript vo verzii 3 takže je očakávateľné predpokladať že podpora pre takúto komunikáciu je na prepracovanej úrovni.

Výhodou Fleshu oproti AJAXu je podpora multimédií. Zatiaľ čo JavaScript je na tom v tom to smere dosť zle Flash vie skoro všetko. Od jednoduchých vektorových a rastrových animácií po 3D animácie a podporu prehrávania zvuku a videa bez nutnosti inštalácie kodekov. Prehrávanie videa je možné oboma smermi a Flash dokáže video nahrávať na server priamo z web kamery nainštalovanej na počítači užívateľa.

Avšak medzi programátormi nieje príliš obľúbený. Programovací model je veľmi odlišný od typických programovacích jazykov. Použitie tejto technológie je spravidla pre užívateľa internetu obťažujúce (reklamné banery a vstupné animácie na stránku) a objavili sa už napríklad aj rozšírenia Firefoxu, ktoré Flash buď dočasne alebo úplne zakazujú. (6)

Na to že je Flash potrebné inštalovať tak je zaujímavé že vo svojom prehliadači ho má aktivovaných 98% užívateľov (údaj z decembra 2006 v USA (31)) avšak len okolo 55% (v závislosti na regióne sa číslo pohybuje 5% oboma smermi) malo v tomto období najnovšiu verziu. V marci toto číslo vzrástlo na viac než 80%. Za takýto rýchly rast vďačíme pravdepodobne popularite služieb na zdieľanie videa, ktoré využívajú najnovšie Fash technológie na zníženie množstva dát potrebných na ukladanie a prenos videí.

3.2.2 Výhody a nevýhody

Flash je najvhodnejším a najrozšírenejším nástrojom ak potrebujeme na stránke multimediálny obsah či už je to video, alebo audio nieje momentálne lepší spôsob ako ho do stránky integrovať ako použitím Flashu. Pokiaľ je cieľom len prehrať video dá sa vložiť do obsahu stránky aj Windows Media Player, no video sa dá v takomto prípade len prehrať a nieje možná žiadna ďalšia interakcia s ostatnými prvkami na stránke, nieje ani možné

pridať logiku správania sa aplikácie. Flash pritom nemusí byť na stránke vôbec viditeľný, ak ide o zvuk, a celé ovládanie môže byt riadené JavaScriptom, aby vzhľad ostal konzistentný so zvyškom HTML obsahu.

Práca s grafikou, vektorovou aj rastrovou, je ďalšou výhodou ktorú má v porovnaní s AJAXom, ktorý nedokáže sám osebe s grafikou pracovať a až v spolupráci s SVG je táto možnosť AJAXovej aplikácii umožnená.

Pozitívnou vlastnosťou je, že keď už Flash na danom systéme beží tak funguje rovnako ako na každom inom. Rôzne správanie sa ako pri JavaScripte nenastáva. Podporované sú všetky relevantné operačné systémy až po Symbian či BeOS a IRIX avšak vo väčšine prípadov len vo verzii 7. Čo je problém ak je vo vyššej verzii spravená dôležitá časť navigácie, napríklad menu.

Flash si vie vytvoriť vlastná spojenie na server (soket) a teda umožňuje skutočnú obojsmernú komunikáciu, kedy môže server poslať informáciu užívateľovi, vtedy keď to potrebuje. Narozdiel od AJAXu kde sa musí JavaScript pravidelne pýtať servera či sa niečo zmenilo.

Flash umožňuje tvorbu pokročilých animácií no aj v najnovšej verzii stále chýba hardvérová podpora pre 3D animácie. Všetko sa renderuje softvérovo čo znižuje výkon a uberá možnosti takto naprogramovaným aplikáciám. No na druhej strane zabezpečuje že všetko čo sa dá vo Flashi na jednom počítači, sa dá aj na inom, bez ohľadu na to či je nainštalované DirectX, alebo OpenGL a či je v systéme inštalovaná dostatočne výkonná grafická karta.

Významným problémom je indexácia v vyhľadávačoch, ktoré možno môžu vedieť pracovať s AJAXovými stránkami, no Flash nieje textový formát, ale skompilovaný binárny súbor. A tam si vyhľadávač nepomôže.

Užívateľovi so zrakovým hendikepom flashovú stránku mu nespracuje rečový

syntetizátor, brailový displej, text si nedokáže zväčšiť ani prepnúť na vysoký kontrast. Od verzie 6 existuje podpora pre takých to užívateľov, ale vývojári na to radi zabúdajú.

Flash je oproti AJAXu potrebné inštalovať, nie je súčasťou žiadneho prehliadača, no aj tak si vybudoval stabilné miesto na internetových stránkach a väčšina používateľov internetu ho má nainštalovaný. Problémom sú firemný zamestnanci bez možnosti inštalácie čohokoľvek ak nemajú postačujúcu verziu tak sa im obsah nezobrazí správne, alebo vôbec.

Preto ak sa nejedná o čisto dizajnovú záležitosť je vhodné vyskúšať ako animácia funguje aspoň o jednu či dve generácie staršej verzii Flashu. A také zásadné elementy akým je napríklad menu by mali mať alternatívu ktorá bude fungovať aj na zariadení na ktorom Flash nieje nainštalovaný, alebo sa dokonca ani nedá nainštalovať

Problém s ktorým sa pri vývoji aplikácií vo Flashi vývojári často stretávajú, je že v hotovom riešení sa zle robia zmeny. Treba si nájsť vhodný pracovný nástroj.

Flash je primárne vektorovo orientovaný a text je tiež reprezentovaný vektorovo jeho kopírovanie (väčšinou) a vyhľadávanie na stránke (ctrl+f) nieje možné, ak je teda účelom stránky textová informácia, alebo práca s textom nieje Flash tou najsprávnejšou voľbou.(6)

3.2.3 Frameworky

Ukážkou čisto na Flashového frameworku sú napríklad OpenLaszlo (46) a Flex od Adobe, ktoré zo vstupu vo formáte XML vytvárajú interaktívne Flashové užívateľské prostredie. V najnovších verziách bola pridaná funkcionalita pre komunikáciu s JavaScriptom a ponúkajú tak podobné možnosti ako knižnica AFLAX, ktorú spomeniem ochvíľu. Oba spomenuté frameworky ponúkajú svoje možnosti aj na strane servera. A prezentačná časť vo Flash plugine potom tvorí prezentačnú vrstvu klasickej trojvrstvovej architektúry.

Kombináciou Flashu a Ajaxu je projekt AFLAX (47). Táto knižnica spája najlepšie vlastnosti AJAXu a Flashu do jedného celku. JavaScriptovým programátorom tak umožňuje využívať možnosti ktoré Flash ponúka bez toho aby museli zmeniť programovací jazyk, s ktorým pracujú. Stránka tak dostane nové možnosti bez toho aby prišla o výhody ktoré plynú z toho že je neje celá spravená vo Flashi.

Abobe momentálne pracuje na alfa verzii projektu Apollo, ktorý má byť platformovo nezávislým prostredím umožňujúcim vývojárom využiť ich existujúce skúsenosti (Flash, Flex, HTML, JavaScript, AJAX) pri tvorbe interaktívnych sieťovo orientovaných aplikácií. Konečná verzia by sa mala objaviť v druhej polovica toho roka pre Windows a OS X a krátko na to aj verzia pre Linux.(48) Bude zaujímave sledovať ako sa Apollo a XAML od MicroSoftu pobijú o svoje miesto na trhu. Aplikácie založené na Apolle nebudú musieť byť spustené v prehliadači a budú sa tváriť ako normálne desktopové aplikácie. Ale s prirodzenou sieťovou podporou. Každá z týchto technológií (Apollo a XAML) bude zaujímavá pre inú skupinu vývojárov, vývojári webaplikácií uplatnia svoje skúsenosti v Apolle a .NEToví v XAML.

Medzi nové možnosti ktoré Apollo ponúka patrí predovšetkým možnosť vytvárať desktopové aplikácie rovnako jednoducho webové. Bez viazanosti na operačný systém bude možné spraviť aplikáciu ktorá bude vyzerať všade rovnako, tak pekne ako sa to vo Flashi dá robiť a pritom sa bude dať tak jednoducho meniť ako HTML.

*2.3 XAML (*eXtensible Application Markup Language)

2.3.1 Základná charakteristika

XAML je deklaratívny na XML založený vektorovo grafický značkovací jazyk slúžiaci na definovanie grafického prostredia, ktorý prišiel s Windows Vista. „A“ v názve pôvodne označovalo Avalon, kódové označenie pre WPF (Windows presentation

Foundation), nové grafické rozhranie vo Windows Vista.

Tento jazyk prišiel ako odpoveď Microsoftu na rozvoj a úspech webaplikácií. Tie aj naďalej budú spúšťané v prehliadači, no ponúknu širšie možnosti užívateľského prostredia, vyššiu bezpečnosť a väčšiu jednoduchosť vývoja. Pomocou jednoducho čitateľného a editovateľného XML formátu je možné definovať užívateľské rozhranie aplikácií oddelene od riadiaceho kódu. Možnosti takto definovaného užívateľského rozhrania nie sú v podstate nijak obmedzené a vyrovnajú sa možnostiam ktoré ponúkajú ostatné jazyky patriace pod .NET.(50)

XAML sa značne využíva v technológiách .NET Frameworku 3.0 a vo WPF, kde je použité ako jazyk na popis užívateľského rozhrania, naviazanie grafických prvkov na zdroj dát (data bindings), volania udalostí a správanie sa objektov. Po nainštalovaní tohto frameworku sú v XAML dizajnované aplikácie spustiteľné vo Windows XP a Windows 2003, no na využitie všetkých ponúkaných možností a hardvérovo urýchlovaných efektov je potrebné ich spúšťať vo Windows Vista. (49)

Vo WF (Windows Workflow Foundation) je XAML používané na popis správania sa systému, niečo na štýl RuleML či BPEL.

XAML bolo vytvorené aby sa užívateľské rozhranie pre Windows písalo jednoduchšie a aby sa dalo použiť aj pre vytváranie web aplikácií. Cieľom bolo aby aplikácie vyzerali tak pekne a mali by sa dať vytvárať tak rýchlo ako ešte nikdy predtým. Mnohé dizajnové prvky ktoré si dizajnéri v minulosti vymysleli si museli nechať pre seba, lebo bolo príliš komplikované niečo také implementovať.

Všetko čo sa dá vytvoriť v XAML môže byť takisto vyjadrené aj použitím tradičnejších .NET jazykov ako C# či Visual Basic .NET. Kľúčovou výhodou technológie XAML je, že nevyžaduje natoľko komplikované prístupy pre použité programovacie nástroje ako spomínané programovacie jazyky, pretože výstupom je obyčajné XML. V procese vývoja softvéru je potom oveľa jednoduchšie, si vymieňať informácie medzi

analytikom, dizajnérom a programátorom a kvôli zmenám, ktoré ktokoľvek z nich do dizajnu dokumentu zapracuje. nieje nutné zasahovať do samotného kódu programu, ktorý má potom na starosti už len funkčnosť.

XAML aplikácie môžu byť spustené samostatne, byť obsahom iframe-u HTML stránky, alebo spustené inou .NETovou aplikáciou. Keď je takáto aplikácia obsahom iframe elementu stránky, JavaScript stránky do ktorej je tento element vložený nemá k nemu žiaden prístup.

Významné vlastnosti ktoré prišli s .NETom a dajú sa pri webaplikáciách na ňom postavených využiť sú Dynamic Language Runtime (DLR) a Common Language Runtime (CLR).(32)

.NET Framework podporuje množstvo rôznych programovacích jazykov. Aby tieto dokázali navzájom spolupracovať využívajú zdielané služby CLR. Časti aplikácie naprogramované v rôznych programovacích jazykoch môžu spolupracovať, využívať spoločné knižnice ako aj stavať na práci v ostatných jazykoch naprogramovaných komponentoch.

V .NET Frameworku 3.0 bol ku CLR pridaný DLR, ktorý umožňuje pracovať s dynamickými jazykmi. Teda jazykmi ktoré môžu za behu meniť svoj zdrojový kód. Tým je umožnená spolupráca medzi dynamickými a statickými jazykmi vzájomne využívať výhody ktoré si ponúkajú vrámci .NET platformy. To umožňuje vývojárom zdieľať kód bez ohľadu na jazyk v ktorom bol naprogramovaný. A takisto si vybrať jazyk nehladiac na prostredie v ktorom bude aplikácia použitá. To všetko pri zachovaní vysokej rýchlosti, ktorú .NET ponúka.

2.3.2 XBAP (XAML Browser Application)

XBAP je skompilovaných niekoľko XAML stránok do celku, ktorý keď je dodaný prehliadaču, tak ten ho zobrazí podobne ako klasickú stránku. Fungujú tlačidlá späť a

vpred, no URL sa nemení a nieje si teda možné uložiť aktuálny stav, ale len odkaz na aplikáciu ako celok.

XBAP je skompilované XAML do binárnej podoby a nieje v ňom z toho dôvodu možné vyhľadávanie vyhľadávačmi. Z rovnakého dôvodu vyplýva aj ďalší problém, keďže je celá stránka skompilovaná, nieje možné stiahnuť si len jedinú stránku, vždy je potrebné stiahnuť súbor ako celok a teda všetky stránky, ktoré stránka/aplikácia obsahuje. Táto technológia je teda skôr vhodná pre tvorbu aplikácií alebo internetových prezentácií na ktoré sa v dnešnej dobe používa Flash ako na klasické internetové stránky.

Okrem toho je XBAP viazané na .NET Framework 3.0 a takáto aplikácia bude spustiteľná len v operačných systémoch Windows 2003, XP a Vista.

2.3.3 Silverlight - WPF/E (Windows Presentation Foundation for Everyvhere)

Silverlight je podmnožinou možností ktoré ponúka XAML a dalo by sa prirovnať ku konkurentovi Flashu svojou platfomovou nezávislosťou (tak v Microsofte volajú podporu pre Windows, aj to len niektoré verzie a MacOS) ako aj možnosťami ktoré ponúka.

Počas dokončovania tejto práce uviedol Microsoft na trh, zatiaľ testovaciu, verziu tohto produktu a zmenil pracovný názov WPF/E na Silverlight. Boli vydané hneď 2 verzie Silverlight 1.0 (beta verzia), ktorý do svojej grafickej plochy vykresluje objekty definované jazykom XAML a pracovať s týmito prvkami je možné pomocou JavaScriptu.

A verzia Silverlight 1.1 (alfa verzia) obsahuje už aj Common Language Runtime 2.0 a Dynamic Language Runtime s podporou nasledovných jazykov: Phyton, JavaScript (ECMAScript 3.0), Visual Basic a Ruby. Ďalšie jazyky budú postupne pribúdať v ďalších verziách Silverlightu. Silu ktorú to tomuto rozšíreniu pridalo zástupca Microsoftu na konferencii v Las Vegas zameranej predovšetkým na prezentáciu Silverlightu prezentoval na šachoch kde 1. hráč bol naprogramovaný v CLR interpretovanom Javasrcirpte a 2. v

klasickom JavaScripte. 1. dokázal prepočítať 1 209 152 ťahov za sekundu a 2. len 651. (60) Čo je 2000 násobne viac. Rýchlosť bude rozhodujúcim faktorom aj v porovnaní s Flashom,

Aplikácia napísaná vo WPF/E sa nedá zobraziť samostatne, ale len ako súčasť HTML stránky a JavaScript má plný prístup k DOMu takejto aplikácie.

Z XAML preberá elementy pre prácu s vektorovou grafikou, obrázkami, videom atď. no napríklad elementy do ktorých by mohol užívateľ písať nepreberá a táto funkcionalita je prenechaná na HTML, alebo na programátora aby si takéto elementy vyrobil.(11)

Aj keď je tvorba WPF/E aplikácie úzko spojená s .NETom nevyžaduje na serveri technológie od Microsoftu a dokáže spolupracovať aj s Apache a PHP. Komunikácia totiž prebieha v štandardných formátoch XML prípadne JSON.(61)

Je ťažké nájsť zásadné funkcie ktoré Silvrelight ponúka na lepšej úrovni ako Flash. Užívateľ si, ak má nainštalované obe rozšírenia, nevšimne rozdiel.

No rozdiely sa nájdu, zatiaľ som sa nestretol s Flashom ktorý by vedel počas prehrávania videa vedel meniť veľkosť zobrazovaného videa a bez straty pozície pokračovať v jeho prehrávaní a vo Flashi je tiež problém s menením rozmerov rastrovým obrázkom.

Z predchádzajúceho vyplýva aj, že použiť video ako textúru bude zatiaľ možné len v Silverlighte.

Rozdiely sa týkajú hlavne programátorov. Prvým je že dáta ktoré prídu zo servera prehliadaču sú vo formáte XML v porovnaní so skompilovaným binárnym súborom v prípade Flashu. Ďalej JavaScript môže pristupovať k DOMu Silverlight elementu a meniť ho. Podpora štandard Windows Media Video (WMV).

Veľmi pokročilá podpora multimédií a práce s videom je prezentovaná na strihaní videa cez Internet Explorer (57 video) na domovskej stránke Silverlightu (1), no takáto aplikácia by sa dala podľa môjho názoru naprogramovať aj vo Flashi (možno až na menenie veľkosti zobrazovaného videa). A pravdepodobne by bol problém s vysokým vyťažením

procesora na strane klienta.

Predovšetkým pre vývojárov pracujúcich s .NETom je najvýznamnejším rozdielom možnosť naprogramovať celú webaplikáciu v programovacom jazyku s ktorým majú svoje skúsenosti a nemusia sa učiť Flash, ActionScript ani JavaScript.

Priame napojenie na služby poskytované Windows Live ako Messenger, Spaces a Contacts sa tiež môže hodiť.

2.3.4 Prípadová štúdia možností XAML

Na stránkach Microsoftu je prípadová štúdia spoločnosti Nike (14) ktorá ukazuje silu ktorú tento nový nástroj na definovanie grafického prostredia má.

Cieľom bolo vytvoriť aplikáciu ktorá by pomáhala predajcom po celom svete mať nielen mať stále aktuálne informácie o produktoch, ale aj v kompletne 3D im ukázať ako si v centrále predstavujú že by mala vyzerať predajná stena (Obr. 3: XAML - Predajná stena).

Obr. 3: XAML - Predajná stena

Celá aplikácia je založené na webservisoch a SOA, ktoré sú pre grafické rozhranie zdrojom informácií. Tak jednoduchá spolupráca 3D grafiky s webovými aplikáciami doteraz nebola možná.

Spôsob akým je XAML navrhnuté umožnilo 5 vývojárom a 3 dizajnérom spolupracovať na tomto produkte a do 3 týždňov odovzdať aplikáciu ktorá sa od bežných programov líši skutočne zaujímavým dizajnom.

Obr. 4: XAML - detail produktu

Na Obr. 4: XAML - detail produktu je v pozadí zoznam farieb v ktorých sa tento produkt vyrába, načítavaný dynamicky podľa aktuálnych informácií

Na tomto príklade je vidieť silu ktorú XAML prináša. Práca s 2D a 3D grafikou je teraz jednoduchým deklaratívnym spôsobom integrovateľná do aplikácií a objektovo orientované API znižuje náročnosť práce s takýmito prvkami. Uľahčuje tak proces vývoja vizuálne zaujímavých aplikácií a otvára vývojárom i dizajnérom nové možnosti, ktorých naprogramovanie nieje tak časovo náročné ako doteraz.

2.3.5 XAML a obvinenie Microsoftu

Názory na XAML sa značne líšia od názoru že to bude nástupca HTML až po podanie na Európsku komisiu že Microsoft chce svojím XAML ovládnuť internet.

Simon Awde(10), predseda Európskej komisie pre súčinné systémy (ECIS) obvinil

Microsoft z plánovaného „ukradnutia HTML“, prevzatia kontroly nad webom a jeho presun preč od otvorených štandardov. Tvrdí že „Vista je prvým krokom v snahe MicroSoftu rozšíriť svoju trhovú dominanciu na internet. Napríklad značkovací jazyk XAML od Microsoftu majúci snahu nahradiť HTML (jazyk aktuálneho štandardu pre publikovanie na internete), je od základov dizajnovaný tak aby bol závislý na Windowse.

S týmto názorom sa nedá úplne súhlasiť, hlavne s časťou vety „od základov dizajnovaný“ keďže XAML je postavený na XML, teda je to obyčajný text, tak ako HTML. Ale interpretácia tohto jazyka je postavená na .NET frameworku 3.0, WPF a ďalších častiach operačného systému Windows Vista. A neexistuje momentálne iná implementácia ako pre tento operačný systém, ktorá by dokázala s XAML plnohodnotne pracovať.

Podľa môjho názoru niečo čo bude platformovo závislé sa nemôže stať jazykom ktorý ovládne internet. Microsoft síce plánuje sprístupniť XAML aplikácie aj pre používateľov operačného systému Windows XP a cez Silverlight (WPF/E) sú takéto aplikácie aspoň v obmedzenej podobe spustiteľné aj v MacOS, no ostatné platformy sa neuvádzajú.

Na miesto náhrady HTML sa technológia XAML resp. Silverlight dá skôr prirovnať k Fleshu. Obe sú vlastnené a patentované súkromnou spoločnosťou a ponúkajú vo web aplikáciách rozšírenie možností užívateľského rozhrania. No Flash má v tomto výhodu je dostupný na širokom množstve platforiem, i keď tie newindowsové zvyčajne meškajú za Windowsovou verziou dátumom vydania, alebo sú niektoré preskočené.

2.3.6 Výhody a nevýhody

Na používane na XAML založených aplikácií je potrebné mať .NET Framework 3.0, ktorý je súčasťou Windows Vista a dodatočná inštalácia je dostupná len pre Windows XP a 2003.

WPF/E (Silverlight) je ako rozšírenie dostupný len pre operačné systémy Windows

Vista a XP s prehliadačmi Internet Explorer a Firefox a pre MacOS na ktorom je podporovaný okrem Firefoxu aj Apple Safari. Microsoft sa rozhodol nevytvoriť verziu pre Linux. Jej vytvorenie prenechal na komunitu vývojárov slobodného softvéru. Miguel de Icaza stojaci za projektom Mono (open-source imlementácia .NET Frameworku) sa vyjadril že Silverlight pre Linux by mohli spraviť do konca roka.(56)

Na podporu svojho nového produktu pripravil pre používateľov Silverlightu zaujímavú službu. 4 GB miesta na svojich serveroch pre videá prehrávané týmto rozšírením.

Aktuálna verzia Visual Studia nemá nástroj ktorým by sa dali „klikacím" spôsobom vytvárať XAML dizajny, ten bude súčasťou až nasledujúcej verzie. (49)

Súčasťou XAML môžu byť okrem dizajnu aj v C# naprogramované reakcie na udalosti vyvolané užívateľom. No ten je vykonaný len v prípade že je toto XAML súčasťou skompilovanej .NET aplikácie. V samostatne spustenej XAML aplikácii je táto časť ignorovaná.(51)

Nato že je XAML nový jazyk založený na XML stojí za povšimnutie, že nieje možné dizajnovanie elementov pomocou kaskádových štýlov, všetko sa dizajnuje priamo v texte. No aj tak je to pokrok z nasledovného C# kódu:

```
TextBox t = new TextBox();
 t.Height = 22;
t.Text = "Môj textbox";
```

sa stane:

```
<TextBox Height="2.2em" >Môj textbox</TextBox>
```

Na pridaní štýlovania na aké sme zvyknutí pri HTML a CSS by mohli v Microsofte popracovať.

2.3.7 Bezpečnosť

Jazyk XAML priamo reprezentuje jednotlivé inštancie objektov a ich funkcionalitu,

tak akoby boli tieto objekty vytvorené klasickým spôsobom v niektorom z programovacích jazykov. Tieto majú prístup k zdrojom počítača limitovanú ďalším komponentom .NET framoworku – Code Access Security. XAML aplikácie sú spúšťané s v internetovej zóne, ktorá má obmedzený, resp. zablokovaný prístup k niektorým systémovým zdrojom. Ak je XAML načítané nejakou aplikáciou, tak má také práva ako aplikácia ktorá ho spustila.

Obmedzenia ktoré sa môžu, no nemusia, na XAML aplikácie spustené v internetovej zóne vzťahovať je okrem iného obmedzený priestor na ukladanie dát na 512KB, zablokovaný prístup do súborového systému, drag&drop, prístup do registrov a pod. Všetko závisí od nastavení systému, čo všetko je zablokované. Prístup k potenciálne zablokovaným zdrojom sa dá samozrejme zistiť a ošetriť.

2.4 XUL (XML-based User-interface Language)

2.4.1 Základná charakteristika

Zjednodušene by sa dalo povedať že je to v podstate XAML od od Mozilly. Avšak s výrazne obmedzenými možnosťami v porovnaní s konkurentom od Microsoftu.

XUL je na XML založený jazyk na popis užívateľského rozhrania umožňujúci popis platformovo nezávislých aplikácií s bohatým užívateľským rozhraním, ktoré môžu no nemusia byť pripojené na internet.

V porovnaní s HTML je XUL zamerané na elementy podstatné pri dizajnovaní aplikácií ako okná, záložky, tlačidlá, hierarchické stromy, atď. Teda prvky ktoré sa vývojári snažia rôznymi spôsobmi simulovať na webstránkach pomocou rozsiahlych JavaScriptových knižníc, sú v XUL už zabudované a dá sa nimi pracovať podobne ako s elementami v HTML. Tieto elementy sú pritom dizajnovo prispôsobené natívnemu vzhľadu konkrétneho operačného systému, no ich dizajn dá prispôsobiť pomocou CSS.

XUL kladie dôraz na dodržiavanie a používanie existujúcich štandardov ako XML 1.0, HTML, CSS 1 a 2, DOM, JavaScript 1.5. A to nie sú všetku ktoré sa dajú využiť, ostáva množstvo ďalších ako XSLT, Xpath, MathML, RDF, Simple Xlinks, Xpointer, XML Base, FIXPtr, XML-RPC, SOAP, WSDL, XBL, HTTP, podpora pre prácu s obrázkami rôznych formátov podporované priamo Geckom, jadro na ktorom je postavené nielen XUL ale aj Mozilla a všetky jej hlavné produkty. Ešte sem patria aj funkcie na prácu s mailami POP3, IMAP, LDAP tie však nepatria pod Gecko, ale sú súčasťou Mozilly.

O sile a možnostiach hovorí napríklad to že celý vzhľad Mozilly je v napísaný v XUL.

Na pridávanie nových prvkov grafického rozhrania slúži štandard XBL. pomocou neho sa dajú vytvárať nové tagy a implementovať k nim funkcionalita.

XUL je dizajnovaný ako platformovo nezávislý, tak aby aplikácie v ňom napísané boli spustiteľné vo všetkých operačných systémoch na ktorých je spustiteľná Mozilla. To je najlepšou vlastnosťou z pohľadu využitia pre webaplikácie ktorú XUL má.

Za vytvorením XUL bola snaha vytvoriť prvotriedne no klasické užívateľské rozhranie, ktoré bude funkčné a ekvivalentne vyzerajúce na všetkých platformách bez toho aby doň bolo potrebné zasahovať. A vyzerať bude tak ako sme zvyknutí, s dizajnom normálnych užívateľských okien, ktorý sa za posledných 10 či 20 rokov nijak výrazne nezmenil. Oproti XAML, ktoré posúva grafické rozhranie aplikácií na nové hranice.

Web programátori so skúsenosťami s DHTML nemajú s naučením sa tohto jazyka problém, lebo je veľmi podobný tomu čo už vedia a vyvíjať aplikácie môžu začať takmer okamžite.

Táto technológia tak ako XAML oddeľuje logickú a prezentačnú časť aplikácie. Ich spájanie je problémom pri vývoji (nielen web aplikácií) keďže ich prípadné zmeny majú už aj v malej firme na starosti rôzny zamestnanci.

Riadenie programovej logiky zabezpečuje XUL, XBL a JavaScript, dizajn má na

starosti CSS a obrázky a jazykové verzie textov sú tiež oddelené.

Inštalovanie takýchto aplikácií je bezproblémové. Mozilla na to vyvinula nástroj ktorý je súčasťou jej produktov ako Firefox či Thunderbird. Tie takúto aplikáciu dokážu po stiahnutí nainštalovať v počítači používateľa avšak inštalácia nieje nutná a tieto aplikácie môžu byť spustené priamo s internetovej adresy.

(13)Ak je ale aplikácia nastavená tak že sa má do systému predsa len nainštalovať nieje nutný klasický inštalačný proces s hľadaním konkrétneho miesta na disku. Výhodou takéhoto inštalovania sú nižšie bezpečnostné obmedzenia a aplikácia má také práva ako program ktorý ju nainštaloval, spravidla to asi bude internetový prehliadač Firefox.

Ak chceme použiť XUL ako aplikačnú platformu je potrebné nainštalovať Gecko Runtime Enviroment, ak ešte v systéme nie je nainštalovaný je potrebné ho stiahnuť spolu s inštalovanou aplikáciou. Sťahovaný balík má 5 až 10 MB v závislosti na platforme. Ak už je v systéme nainštalovaná dostatočná verzia stačí nainštalovať len aplikáciu.

Podobne ako Firefox či mailový klient Thunderbird aj ostatné aplikácie naprogramované pomocou XUL sú jednoducho rozšíriteľné.

Ambíciou XUL je nahradiť množstvo klientských rozhraní klient-server aplikácií, ktoré si každá spoločnosť vyvíjajúca takúto aplikáciu musela sama naprogramovať. Teraz je tu štandardizovaný jazyk na popis takéhoto užívateľského rozhrania s licenciou slobodného softvéru.

(12)XUL zmazáva rozdiely medzi desktopovou aplikáciou a internetovým prehliadačom, pretože je pevne udomácnená v oboch svetoch. Web aplikácie budú ťažiť z prechodu na XUL, získajú tým nové možnosti užívateľského rozhrania, rovnaký vzhľad na všetkých platformách a prirodzený prístup k zdrojom ako sú zdieľané knižnice a súborový systém na počítači užívateľa.

Ako aj ak je potrebný prechod web aplikácie na desktopovú. Pri hľadaní technológie ktorá by umožnila jednoducho vytvoriť aplikáciu spustiteľnú na viacerých platformách.

Alebo pri začlenení novej funkcionality do prehliadača či mailového klienta je XUL vhodné riešenie.

V roku 2005 OpenXUL Alliance napočítala 21 samostatných komerčných implementácií založených na XUL.

2.4.2 Výhody a nevýhody

Medzi výhody na prvé miesto patrí kompatibilita so štandardami, pretože okrem AJAXu ktorý je z nich odvodený, toto je jediná technológia ktorú v tomto dokumente spomínam a ktorá sa štandardov svedomito drží.

XUL prináša jednotné užívateľské rozhranie na všetky platformy.

Pre užívateľov prehliadača Firefox a ostatných na Mozille založených je výhodou aj to že na spustenie v XUL napísaných aplikácií nemusia robiť nič len si otvoriť danú stránku. No ak používajú iný prehliadač, nastáva problém. V prípade Internet Explorera som dlho považoval túto technológiu sa nevyužiteľnú. No existuje riešenie (22) ktoré umožní spúšťanie XUL aplikácií aj v tomto prehliadači. No bolo by vhodné vytvoriť klasický inštalačný balík ktorý spraví všetko potrebné. Skúsený užívateľ pokiaľ zistí že takéto riešenie existuje nemá problém podľa návodu postupovať a vyriešiť tento problém. No bežný užívateľ to okamžite vzdá.

Problémom XUL bolo že neexistovali nástroje na v ktorých by sa dali takéto aplikácie jednoducho vyvíjať, no už to problémom nieje. (33)

2.5 Java Applety

2.5.1 Základná charakteristika

Java applet je program naprogramovaný v programovacom jazyku Java, ktorý môže

byť vložený do HTML stránky, podobne ako sa do stránky vkladajú obrázky. Pokiaľ návštevník stránky má prehliadač podporujúci Java technológiu, kód appletu je posunutý operačnému systému a vykonaný v Java Virtual Maschine (JVM).(52)

Java applety boli prvý plnohodnotný programovací jazyk použiteľný v internetových prehliadačoch. Predstavený bol v roku 1995.

Vďaka tomu že program napísaný v Jave, presnejšie ich bytecode, je spracovávaný JVM, je takýto program platformovo nezávislý, no JVM musí byť na počítači užívateľa nainštalovaná. Cieľom spoločnosti Sun, ako tvorcu tejto technológie bolo aby programy napísané v Jave boli spustiteľné kdekoľvek a preto JVM existuje pre každú „štandardnú" platformu. Existujú aj nástroje (applet2app) na konverziu Java appletov na klasické v jazyku Java napísané programy, ktoré je po konverzii možné spúšťať bez internetového prehliadača a pripojenia na internet.

Keďže applety sa spúšťajú vo virtuálnom stroji, ktorý má značné bezpečnostné obmedzenia vzhľadom k materskému operačnému systému, je problém spúšťať aj neoverené applety. No pokiaľ to užívateľ dovolí môžu byť takémuto programu sprístupnené všetky zdroje ktoré má systém k dispozícii. Kým takýto applet beží v nedôveryhodnom móde má výrazne obmedzený prístup k systémovým prostriedkom, no aspoň k ním má potenciálny prístup, narozdiel od Javascriptu a teda Ajaxu.

Tak ako Flash aj Java applety potrebujú aby bol v systéme nainštalovaný program ktorý ich dokáže spustiť, počet užívateľov ktorí majú JVM v počítači inštalovaný je vyše 87% (31) oproti Flashu je tak nižší o viac než 10 percentuálnych bodov. A pred spustením prvej aplikácie je nutné aby sa spustil virtuálny stroj v ktorom bude následne spustený applet, tento čas môže byť dosť dlhý.

Užívateľské rozhranie Java appletov nie celkom dizajnovo ladí so zvyškom HTML stránky. Na grafickom prostredí týchto aplikácií sa za niekoľko rokov nič nezmenilo a

pokiaľ dizajnu nieje pri vývoji venovaný dostatočný dôraz, tak aplikácia nepôsobí moderným dojmom.

V minulosti boli Java applety najsilnejším nástrojom a jediným plnohodnotným programovacím jazykom ktorý sa dal použiť na strane klienta na vytvorenie interaktívneho užívateľského rozhrania v prehliadači, no postupom času ich funkcie dokázal nahradiť JavaScript a Flash, ktoré sú na tieto účely využívané oveľa častejšie. Dôvodom je že užívateľské rozhranie v Flashi je oproti Javovému značne krajšie. Flash bol od počiatku vyvíjaný ako nástroj na tvorbu pekného multimediálneho obsahu, narozdiel od Javy ktorá je vo svojej podstate programovacím jazykom a za posledné roky jej vývojári nedali toľko čo sa investovalo v Macromedii a Adobe do Flashu.

2.5.2 Výhody a nevýhody

Najväčšou výhodou Javy oproti ostatným technológiám je množstvo knižníc so širokým záberom funkcií.(36)

Komunikáciu medzi appletom a prehliadačom sprostredkúva Javascript. Prístupné sú všetky public metódy appletu (document.meno_appletu.public_metoda***).***

Pomalý štart Java appletov je jedným z hlavných dôvodov prečo sa nepresadili pri tvorbe interaktívneho obsahu na internetových stránkach. Ako aj viacero chýb ktoré sa prejavovali na všetkých platformách (35)

2.6 Štandardy vo vývoji

2.6.1 Web Applications 1.0 (HTML5)

Iniciatíva WHATWG (Web Hypertext Application Technology Working Group). Cieľom je aby táto špecifikácia predstavovala vlastnosti HTML a DOMu ktoré majú uľahčiť vytváranie webaplikácií. Pridáva kontextové menu, grafickú plochu do ktorej sa dá priamo pristupovať, inline popup okná a serveru posielané udalosti.(16)

Tento dokument je zatiaľ vo vývoji a pri rýchlosti akou sa štandardy dostávajú nielen do najrozšírenejšieho prehliadača na svete je ťažké odhadnúť kedy bude reálne použiteľný v praxi. No už sa objavili aj plány na HTML6 ktoré má byť kompatibilné s ISO štandardom OOXML a pomocou XSLT by mali byť jednoducho konvertovateľné stránky napísané v HTML4. (53)

3 Porovnanie spomenutých technológií

3.1 Bezpečnosť

Najväčšie problémy ohľadne bezpečnosti zo spomínaných technológií sa týkajú JavaScriptu a teda AJAXu. Nieje problémom to že celý zdrojový kód takejto aplikácie je čitateľný v akomkoľvek textovom editore a teda sa dá jednoducho a presne zistiť, ako vytvára požiadavky odosielané na server a netýka sa to len AJAXu, ale i XUL a XAML (SilverLight). Problémom je, že komunikácia medzi prehliadačom a serverom prebieha cez nešifrovaný protokol HTTP. Teda ak niekto odpočúva takúto komunikáciu nemusí vynaložiť žiadnu námahu na jej dešifrovanie, stačí aby pochopil čoho sa jednotlivé požiadavky a odpovede na ne týkajú.

Tomu to problému sa budem viac venovať v kapitole kapitole 6.5.

Z tohto pohľadu sú ostatné technológie v poriadku.

3.2 Rýchlosť vývoja

V tejto kapitole som chcel porovnať jednotlivé technológie z hľadiska rýchlosti vývoja ako aj rýchlosti akou sa dajú jednotlivé technológie naučiť.

Problémom je že pre AJAX existuje množstvo frameworkov, ktoré mnoho činností do značnej miery zjednodušujú. Pri Flashi je situácia podobná, no frameworkov nieje až tak veľa. Pre XUL existuje viacero programov, ktoré zjednodušujú vývoj takýchto aplikácií. A XAML a Silverlight môžu byť vyvíjané vo viacerých programovacích jazykoch súčasne.

Na to aby som mohol porovnať rýchlosť vývoja aplikácií v jednotlivých technológiách, musel by som byť skúseným programátorom v každej spomínanej

technológii. Ideálne aby som vedel pracovať aj viacerými alternatívami, ktoré daná technológia ponúka. Aby som mohol tvrdiť že riešenie, ktoré som použil je najlepší spôsob ako sa vysporiadať s nejakým problémom. A teda malo vôbec nejaký zmysel sa k tomu vyjadrovať. Z tohto dôvodu sa nedovážim porovnávať jednotlivé technológie podľa toho kritéria.

3.3 Platformová nezávislosť

3.3.1 Z pohľadu operačného systému

Najhoršie z tohto porovnania vychádza XAML, ktoré plnohodnotne funguje len na Windows Vista. Pre Windows XP a Windows 2003 po nainštalovaní .NET Frameworku 3.0 budú spustiteľné aj aplikácie vyžívajúce XAML no nie so všetkými možnosťami ktoré táto technológia ponúka (napr. hardvérové urýchľovanie 3D efektov). Avšak existuje okresaná verzia XAML ktorá je dostupná pre Windows XP, 2003 a MacOS X pod označením Silverlight a časom by mala vzniknúť aj verzia pre Linux a Windows 2000, ktorá ponúka dostatočne zaujímavé možnosti.

V prípade plnohodnotného XAML úspech a rozvoj závisí od toho ako sa bude dariť najnovšiemu operačnému systému od Microsoftu na trhu, v ktorom ako jedinom je XAML plnohodnotne využiteľné. Podľa štatistiky (25) z marca 2007 viac než 15% užívateľov internetu (Windows 98 a 2000, MacOS, Linux a ostatné operačné systémy) nebude mať ani len možnosť nainštalovať si .NET Framework 3.0 a len necelé 2%-tá (Windows Vista) si takouto technológiou vytvorenú aplikáciu spustia bez nutnosti stiahnuť si zo stránok Microsoftu inštalačný balík o veľkosti 50MB. Reálny počet užívateľov s možnosťou okamžite využiť možnosti ponúkané XAML bude podľa môjho vlastného odhadu hlboko pod 10% no časom sa bude každopádne zvyšovať.

Lepšie je na tom Silverlight, ktorý si momentálne nebude môcť nainštalovať viac

než 11,5% (Windows 98 a 2000, Linux, staršie verzie MacOS a ostatné operačné systémy) užívateľov a i keď bez inštalácie sa nedá využiť na žiadnom operačnom systéme inštalácia je nanajvýš jednoduchá a veľkosť sťahovaného súboru je výrazne menšia 1 resp. 4 MB v závislosti od verzie. No aj tak kým užívatelia nebudú mať Silverlight nainštalovaný nedá sa bez problémov využívať. Na druhej strane kým sa na stránkach nebude používať, užívatelia nebudú nútení si ho inštalovať.

Všetky ostatné technológie sa dajú plnohodnotne využívať na každom operačnom systéme s výnimkou mobilných telefónov kde je podpora pre spomínané technológie pomerne slabá.

V niektorých mobilných telefónoch sa dajú spúšťať Flash aplikácie no tie skôr fungujú v samostatnom Flash prehrávači ako priamo v internetovom prehliadači.

Opera má vo svojom prehliadači pre mobilné telefóny Opera Mobile, nie Opera Mini, podporu pre AJAX. No keďže má v takomto prípade prehliadač veľmi malý displej mala by na to byť aplikácia prispôsobená.

3.3.2 Z pohľadu prehliadača

V tomto prípade neberiem do úvahy XAML, pretože ktorýkoľvek prehliadač, ktorý stiahne súbor s príponou .xaml a pokúsi sa ho spustiť, tak pokiaľ v systéme bude nainštalovaný .NET Framework 3.0 spustí danú aplikáciu. V tomto prípade platí čo bolo povedané v predchádzajúcej kapitole.

Najhoršie z tohto porovnania vyhádza XUL. Firefox je síce dostupný pre každú platformu no prirodzená podpora je len v ňom a v ostatných prehliadačoch ktoré sú odvodené od Mozilly. A ako už bolo spomenuté v kapitole 3.4.2 je možné sprístupniť takéto aplikácie aj užívateľom Internet Explorera, no inštaláciu by bolo vhodné zjednodušiť. Chcieť od bežného užívateľa by si stiahol zbalený program, rozbalil ho niekam na svoj disk,

skopíroval do toho istého adresára ďalšie 2 súbory, ktoré si musí tiež stiahnuť z internetu a jeden z nich spustil je bohužiaľ viac než je možné od neho očakávať.

Vyjadrené v číslach opäť podľa štatistiky (25) z marca 2007 bude takáto aplikácia okamžite, bez nutnosti čokoľvek inštalovať, funkčná pre 33% užívateľov (Firefox a Mozilla). A ďalších takmer 57% (Internet Explorer 6 a 7) má možnosť postupovať podľa spomenutého návodu (22), no kým nebude inštalácia jednoduchšia nedá sa predpokladať že to čo i len malá časť z nich spraví, hlavne pokiaľ na to nemá dôvod.

Po XUL je na tom najhoršie Silverlight. Okrem toho že Silverlight aplikácie fungujú len na niektorých operačných systémoch, fungujú aj len v niektorých prehliadačoch: Internet Explorer 6 a 7, Firefox 1.5 a 2.0 a Safari 2.0. Do budúcna je plánovaná aj Opera no prehliadačov existujú desiatky, a zatiaľ podporované pokrývajú len niečo viac než 90% trhu (25). Avšak toto číslo, vyjadrujúce množstvo podporovaných prehliadačov, treba ešte o niečo znížiť, kvôli užívateľom používajúcim podporovaný prehliadač na nepodporovanom operačnom systéme, podľa použitej štatistiky nieje možné presne určiť množstvo takých to užívateľov, no ich počet bude niekde medzi 80 a90%.

Flash má deklarovanú podporu len na obmedzenej skupine prehliadačov: Internet Explorer 6 a 7, Firefox 2, Netscape 7 a 8, Safari 2.0 a Opera 9.1 (9) no dá sa doinštalovať do viacerých ktoré v tomto zozname nie sú v marci 2007 ho malo nainštalovaný viac než 98% užívateľov. (31)

AJAX keď je dostatočne dobre naprogramovaný a rovnako dobre otestovaný mal by fungovať takmer všade. Ako jediný je len rozšírením HTML a táto vlastnosť sa dá využiť. Ak totiž daný prehliadač nepodporuje JavaScript, alebo ho nepodporuje v dostatočnej miere dá sa na miesto ošetrovania užívateľovej aktivity JavaScriptom poslať požiadavka na server a následne prekresliť celá stránka. V prípade že niektoré problematické časti nie sú takýmto spôsobom ošetrené, niektorí užívatelia môžu mať niektoré funkcie trvalo

neprístupné. Avšak narozdiel od ostatných technológií je zachovaná aspoň čiastočná funkcionalita. V januári 2007 malo prehliadač s (aspoň čiastočnou) podporou JavaScriptu 94% užívateľov.

Je zaujímavé, že podpora pre Flash je o 4 percentuálne body vyššia ako pre Javascript. Nemyslím si že by 4 % užívateľov internetu nemalo aktivovaný JavaScript a zároveň by boli schopní spúšťať Flashové aplikácie. Vysvetlení pre tento nepomer by mohlo byť viacero. Iná reprezentatívna vzorka, zavádzanie zo strany Adobe, alebo v štatistike na w3chools.com (25) neodpočítali návštevy spôsobené vyhľadávačmi.

3.4 Znevýhodnené skupiny užívateľov

Internet je pre všetkých a preto by informácie na ňom zverejnené mali byť dostupné naozaj všetkým. Či už návštevníkom s poškodením zraku, ale aj tím čo nemajú nainštalované alebo zapnuté všetky potrebné rozšírenia (Flash, Javascript,...). Je jasné že v takomto prípade nieje možné poskytnúť užívateľovi všetky možnosti ktoré stránka či aplikácia ponúka, no aspoň základná informačná hodnota by mala byť dostupná pre každého.

Touto problematikou sa zoberajú 2 rozsiahle dokumenty od W3C Roadmap for Accessible Rich Internet Applications (18) a Web Content Accessibility Guidelines 1.0 (54) Druhý spomínaný dokument požaduje aby bola stránka prístupná aj používateľom s vypnutým JavaScriptom. Momentálne je vo vývoji verzia 2.0, ktorá toto obmedzenie vypustila kvôli rozvoju webaplikácií.(59)

Je potešujúce ma že k tejto problematike sa postavili v Mozille, Microsofte i Adobe zodpovedne a súčasťou dokumentácie je vo všetkých prípadoch aj súbor odporúčaní ako robiť takéto aplikácie prístupné pre znevýhodnených užívateľov.(37)(38)(58)

Pokiaľ by sa programátori držali všetkých týchto pravidiel a odporúčaní, tak by

akoukoľvek technológiou vytvorená aplikácia bola dostupná pre kohokoľvek. Zodpovednosť je teda na programátoroch aby užívateľov neobmedzovali.

3.5 Hranice jednotlivých technológií a ich najvhodnejšie uplatnenie

3.5.1 AJAX

Je najslabšou zo spomínaných technológií. Chýbajúca podpora pre prácu so zvukom, vektorovou i rastrovou grafikou a videom nieje pri mnohých druhoch aplikácií problémom.

Závažnejšie je, že funkčnosť vo všetkých prehliadačoch nemusí byť stopercentná, kvôli rôznej implementácii Javascriptu a DOMu. A pri veľkom množstve JavaScriptového kódu začne byť prehliadač pomalý.

Kým sa nepracuje s dôvernými dátami nezáleží ani na tom že dáta nedokážu prúdiť cez nezabezpečený komunikačný kanál.

Napriek tomu všetkému sa zo všetkých spomínaných technológií používa najčastejšie, pre svoju spätnú kompatibilitu s HTML a z toho vyplývajúce výhody.

Nedostatky týkajúce sa vektorovej grafiky sa dajú vyriešiť použitím SVG, no malý počet užívateľov internetu používajúcich prehliadač schopný tento štandard využiť a rôzna úroveň podpory tohoto štandardu v prehliadačoch je dôvodom prečo sa zatiaľ na internete neujal v takej miere ako to jeho možnosti umožňujú.

Rozšírenie AJAXu o podporu multimédií ako aj vektorovej grafiky umožňuje s využitím Flashu knižnica Aflax (spomínaná v kapitole 3.2.3). V prípade Silverlightu takáto knižnica nieje nutná, keďže JavaScript má k jeho DOMu priamy prístup.

AJAX svoje uplatnenie nachádza predovšetkým (34) pri práci s formulármi

(automatické ukladanie, dopĺňanie, obmedzovanie možností na základe zadaných údajov), hlbokých hierarchických stromoch, priamej komunikácii medzi užívateľmi, pri hlasovaní, triedení a zoraďovaní údajov a pod. Teda na stránkach ktorá chcú užívateľovi spríjemniť ich používanie minimalizovaním počtu načítaní.

Je to všeobecnosti najvhodnejšia technológia na tvorbu aplikácií, ktoré by mohli bez väčších problémov existovať aj bez JavaScriptu, no ak je použitý tak návštevník takejto stránky nedostane na každú akciu ktorú vykoná načítanie novej stránky, ale priamu odpoveď na svoju požiadavku.

3.5.2 FLASH

Žiaden z problémov AJAXu sa Flashu netýka. Až na problém s rýchlosťou, ktorý však v tomto prípade nezávisí od množstva kódu. Flash podporuje množstvo náročných funkcií, ktoré nie sú vykonávané tak efektívne ako v prípade iných programovacích jazykov.

No má zopár vlastných problémov, je to skompilovaný binárny súbor v ktorom nedokážu vyhľadávať vyhľadávače, ani užívatelia pomocou ctrl+f, väčšina textov sa nedá kopírovať.

Flash vznikol ako nástroj na tvorbu graficky zaujímavých efektov a prepracoval sa na najrozšírenejší nástrojom pomocou ktorého sa dá pracovať s multimediálnym obsahom na stránke.

Najčastejšie nachádza svoje uplatnenie (6) pri prehrávaní i nahrávaní zvuku a videa, komplexných animáciách, vďaka tomu že si dokáže vytvoriť obojsmerné spojenie so serverom pri hrách s viacerými hráčmi.

3.5.3 XAML, Silverlight

Ide o novú technológiu ktorá zatiaľ neukázala svoje technologické nedostatky. Hlavný nedostatok ktorý na nej je sa týka marketingu a to skutočná platformová nezávislosť.

Negatívom nie tejto technológie, ale pre túto technológiu je, že zatiaľ neexistuje vývojové prostredie na tvorbu dizajnu na operačnom systéme MacOS. Pretože množstvo grafikov používa práve túto platformu na svoju prácu.

Svoje uplatnenie nájde v tvorbe úplne nového typu aplikácií, no dokáže aj všetko čo ostatné a omnoho viac.

3.5.4 XUL

Nevýhodou XUL je že nemá podporu multimédií ani práce s grafokou.

Hlavné zameranie je na tvorbu užívateľského rozhrania funkčného na každej platoforme.

3.5.5 Java applety

Množstvo vytvorených aplikácií a knižníc tejto platforme zabezpečí že sa bude ešte dlho používať.

Kvôli pomalému štartu sa nepoužíva na tvorbu malých aplikácií, lebo čas spúšťania by mohol byť porovnateľný s časom ako dlho bude daná aplikácia spustená.

Svoje uplatnenie má pri platformovo nezávislých aplikáciách, ktoré využívajú v Jave vytvorené komponenty, nepotrebujú graficky tak zaujímavé prostredie ako ponúka Silverlight, potrebujú vyšší výkon ako ponúka Flash, alebo na strane klienta vyžadujú väčší stupeň interaktivity ako ponáka XUL.

4 Aktuálny stav používania technológií

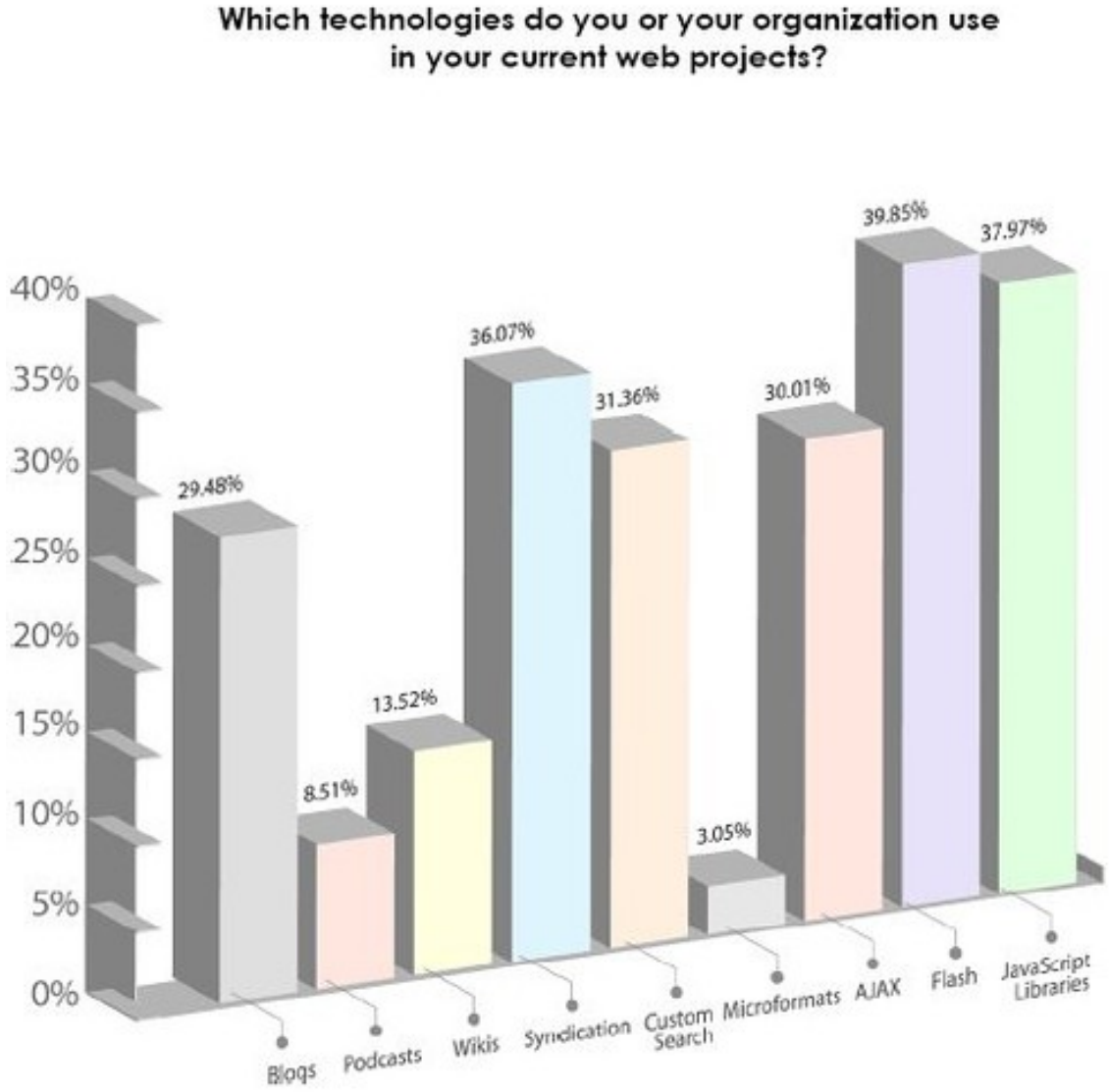
4.1 Technológie používané pri tvorbe internetových stránok

Podľa prieskumu vykonaného na vzorke 1,813,471 internetových stránok zobrazuje zastúpenie technológií ktoré pri ich tvorbe boli použité.

Technológia	Počet stránok	Percentá
JavaScript	1,083,999	59.77%
Frame	252,016	13.90%
CSS	979,251	54.00%
Java	22,171	1.22%
IFrame	195,195	10.76%
GIF	1,136,176	62.65%
JPG	998,450	55.06%
PNG	175,569	9.68%
Flash/Shockwave	231,566	12.77%

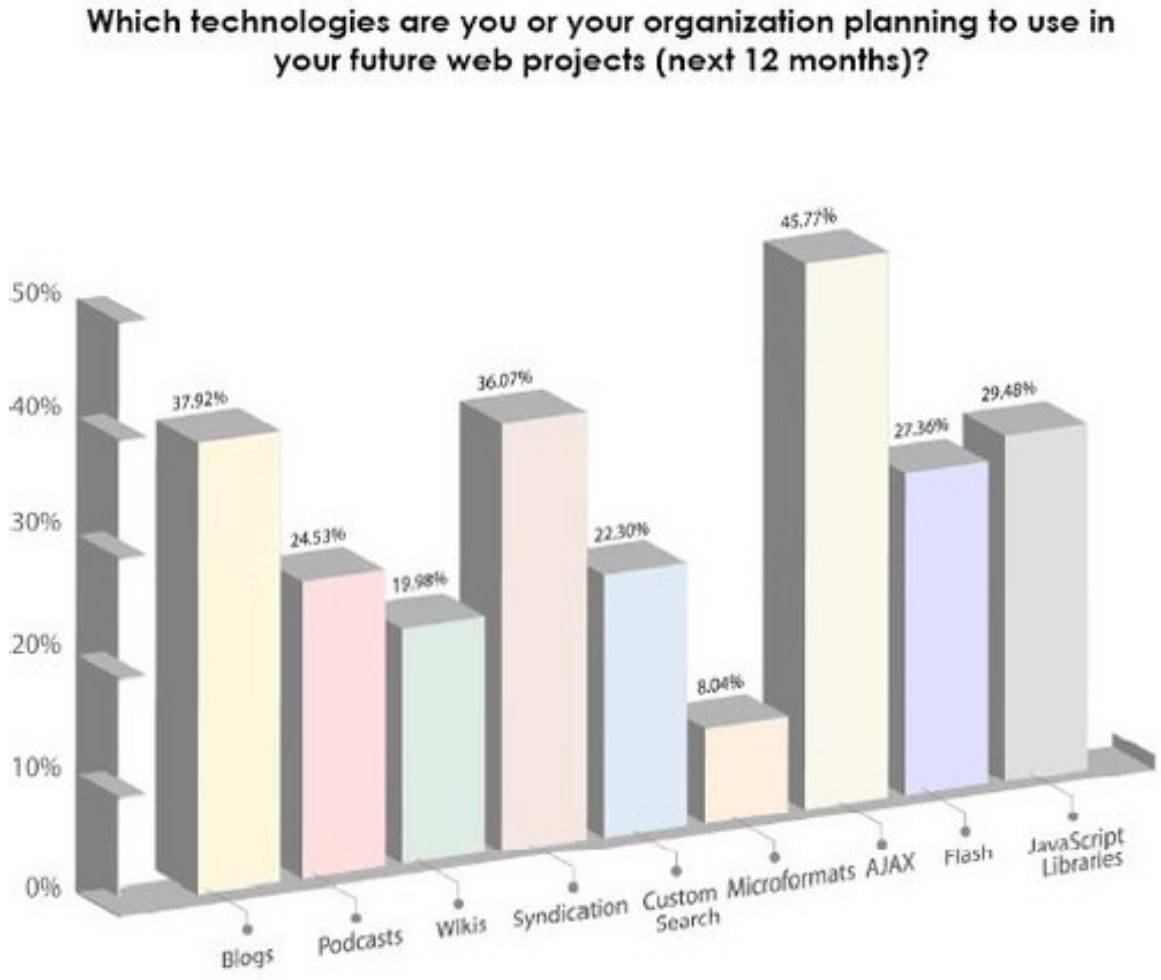
(17)

4.2 Prieskum medzi 5000 americkými vývojármi, aké technológie sa využívajú v ich firmách pri tvorbe internetových projektov.



Obr. 5: Aké technológie sa používajú pri tvorbe internetových porjektov

4.3 A aké plánujú využívať v budúcnosti



Obr. 6: Aké technológie planú využiť v nasledujúcom roku

Zaujímavé na týchto 2 grafoch je porovnanie AJAXu a Fleshu, zatiaľčo Flash vo svojich projektoch využíva takmer 40% spoločností a AJAX 30% tak v budúcom roku sa tento pomer má obrátiť. A poprvýkrát by sa tak mal AJAX dostať na čelo tohto porovnania.

zdroj 21 (z komplexného pohľadu na Stav vývoja web aplikácií v roku 2006(20))

5 Ako na najčastejšie problémy AJAXu

V tejto kapitole je rozobratých niekoľko hlavných problémov súvisiacich predovšetkým s AJAXovými aplikáciami a priblížené niektoré možnosti ich riešenia.

Pri použití niektorého z množstva dostupných frameworkov je veľmi pravdepodobné, že tento bude mať mnohé z problémov, ktorých riešenia sú v tejto kapitole navrhnuté už implementované. Tak táto časť je predovšetkým pre tých ktorý sa rozhodli postaviť AJAXovú aplikáciu od základov, alebo potrebujú rozšíriť existujú aplikáciu o AJAXové prvky.

5.1 Uchovanie stavu

On-line počítačový slovník (19) slovo stav vysvetľuje nasledovne „To ako niečo je; jeho nastavenia, vlastnosti, podmienky alebo informačný obsah. Stav systému je spravidla dočasný (t.j. mení sa v čase) a prchavý (t.j. bude stratený, alebo vynulovaný do nejakého počiatočného stavu).“

Všetky softvérové i hardvérové prvky majú svoj stav, od aplikácií, cez operačný systém až po sieťové vrstvy. Za stav považujeme bod v nejakom priestore všetkých možných stavov. Prvok môže svoj stav zmeniť s jedného stavu na iný v čase keď zaznamená nejaký druh udalosti. Takouto udalosťou môže byť sieťová správa, vypršanie časovača, alebo správa od aplikácie. Prvky ktoré nemajú stav, teda také že neexistuje spúšťač ktorý by mohol spôsobiť zmenu stavu sa nazývajú bezstavové. Najzaujímavejšie, čiastočne personalizované, prvky majú nejaký druh stavu, ktorý im umožňuje poskytovať personalizovanú informáciu keď dôjde k interakcii s užívateľom. Toto sa týka predovšetkým týchto druhov stavov:

1. Stav aplikácie, všeobecný stav samotnej aplikácie
2. stav zdrojov, ide predovšetkým o stav vo vzťahu k identifikátoru určujúcemu zdroj (URI). V kontexte internetových stránok stav aplikácie zodpovedá stavu jej zdrojov, teda stav týchto zdrojov je stavom aplikácie.
3. stav vo vzťahu k užívateľovi alebo sedeniu (session), ktorý môže spôsobiť že aplikácia sa rôznym užívateľom správa rôzne (23)

Dôležitou vlastnosťou systému je aby dokázal uchovať resp. obnoviť svoj stav a pokračovať v práci tam kde prestal. Pri internetových stránkach si na to užívatelia zvykli a posielanie linkov a pridávanie si stránok k obľúbeným (bookmarky) je tiež jeden zo spôsobov ako uchovať aktuálny stav pre budúce použitie.

Pri statických stránkach, ako aj pri stránkach vytváraných na serveri, je každá takáto stránka jednoznačne identifikovaná svojou adresou URL a obsahom cookies, či už na strane klienta alebo servera.

Pri webaplikáciách sa dostávame do stretu medzi aplikáciami a internetovými stránkami. Na jednej strane sú klasické internetové stránky pri ktorých si vieme zapamätať link a kedykoľvek sa na dané miesto vrátiť a pokračovať. Na strane druhej desktopové aplikácie, v ktorých sa dá pokračovať zakaždým len od konkrétneho nami uloženého stavu, ktorý zodpovedá uloženému niekde na disku uloženému súboru. Avšak pri webových aplikáciách do toho vstupuje ešte ďalší faktor ktorým je prerušenie spojenia a následné zmarenie všetkej vykonanej práce.

Našťastie existuje viacero spôsobov ako stav uchovať. Tieto sa dajú rozdeliť do 2 hlavných kategórií. Buď sú v prehliadači uložené všetky potrebné premenné a ich hodnoty potrebné na správne zobrazenie stránky, alebo si prehliadač pamätá len identifikátor podľa ktorého je na serveri poskladaný stav danej aplikácie.

5.1.1 Uloženie stavu na strane klienta

Výhody tohto prístupu sú:

- Ak je v prehliadači je kompletná informácia o stave, a JavaScript ju vie spracovávať, nielen posielať serveru, môže to znížiť množstvo požiadaviek posielaných na server.
- Všetky stavy boli uložené v prehliadači čo ich umožňuje využiť pri ošetrovaní funkcionality tlačidla „späť“.
- Nenastane vypršanie session.

Samozrejme má táto metóda aj svoje nevýhody:

- Množstvo prenesených dát pri každej požiadavke. Spôsobené neustálym posielaním všetkých dát, ktoré sú na obnovenie stavu potrebné.
- Možné väčšie vyťaženie procesora na strane klienta a následné spomalenie celej aplikácie.

Najjednoduchším spôsobom ako si uchovávať stav na strane klienta je poslať zo servera serializovaný objekt, alebo pole so všetkými potrebnými hodnotami.

5.1.1.1 Uloženie stavu v URL.

JavaScript môže meniť časť URL za '#' a v tomto priestore je teda možné si ukladať dáta ku ktorým môže neskôr pristupovať a pracovať s nimi.

Jednalo by sa o najlepší spôsob, pretože by to vyriešilo problémy s posielaním linkov aj ukladaním si stránok medzi obľúbené. No v Internet Exploreri je dĺžka URL obmedzená na 2083 znakov (28), čo je pre komplexnejšie aplikácie málo. Na druhej strane to často krát môže stačiť.

5.1.1.2 Uloženie stavu v cookies.

Tento priestor je limitovaný 4 kilobajtami na 1 cookie a na 1 doménu môže byť nanajvýš 20 cookies. to je spolu 80 KB informácií ktoré sa dajú takýmto spôsobom uchovávať.

Treba myslieť na to že takéto ukladanie dát nieje bezpečné, pokiaľ tieto dáta nie sú šifrované. A na server sa posielajú s každou jednou požiadavkou.

Ďalším problémom ktorý treba očakávať je že užívateľ môže cookies kedykoľvek vymazať a na ich existenciu sa teda nedá 100%-ne spoliehať. No ak užívateľ niečo zmaže nemôže sa očakávať že všetko mu bude bezchybne fungovať.

Môže sa stať aj to že užívateľ nemá cookies zapnutéa teda sa mu takéto dáta nemajú kde uložiť.

5.1.1.3 V obsahu stránky

Do obsahu stránky je pridaný skrytý element v ktorom sú potrebné informácie uchovávané no v tomto prípade nieje žiadna informácia zapamätaná po zavretí okna prehliadača. No dá sa využiť na ošetrenie tlačidla „späť“.

5.1.1.4 V súborovom systéme užívateľovho počítača

Tento spôsob je dostupný v Internet Exploreri pomocou ActiveX a vo Firefoxe pomocou XUL dá sa to i v Safari, no keďže nejde o štandardný postup nieje vhodné ho používať.

5.1.1.5 Vo Flashi

Adobe Flash takisto poskytuje možnosť ako si ukladať informácie v počítači používateľa. Flash môže zapisovať a čítať údaje zo zdieľaného objektu, ktorý ostane v

počítači uchovaný aj keď je stránka už zatvorená. (27)

Flash plugin môže uchovávať na lokálnom počítači dáta do veľkosti 100KB. Pri pokuse uchovať väčšie množstvo dát sa zobrazí dialógové okno, ktoré sa užívateľa spýta či povoliť alebo zakázať uložiť takéto množstvo dát. O úspešnosti tejto akcie je samozrejme program ktorý o to žiadal informovaný. Veľkosť miesta ktoré je možné na počítači uložiť nieje možné pomocou ActionScriptu ovplyvniť, táto hodnota je len na čítanie.

5.1.2 Na strane servera

Tento prístup má blízko k tradičným internetovým stránkam. Užívateľ je identifikovaný pomocou session na základe ktorej sa z databázy načítajú údaje potrebné na poslanie odpovede. A následne sa do databázy nové, zmenené dáta uložia. Vhodné je ak je užívateľ prihlásený môže tak v práci pokračovať aj po dlhšom časovom intervale ako je čas na vypršanie session.

Pri takomto spôsobe riešenia uchovávania stavu sú samozrejme obnoviteľné len tie časti práce ktorých zmeny boli oznámené na serveru.

5.2 Ošetrenie stlačenia tlačidla „späť“

Ak nieje žiadúce aby na AJAXovej stránke prehliadač po stlačení tlačidla „späť“ skočil na predchádzajúcu stránku, ale vrátil sa o jeden krok späť z pohľadu JavaScriptu je potrebné to dať nejakým spôsobom prehliadaču vedieť.

Existujú 2 zaužívané spôsoby ako na to. Fungujú tak že v každom kroku, ktorý má byť zapamätaný a má sa dať do neho vrátiť po stlačení tlačidla „späť“, musí byť vykonaná nejaká akcia.

V prvom prípade je potrebné mať na stránke iframe, a spomínanou akciou je zmena obsahu iframe-u. Ten síce nemusí byť skrytý, no aby nezavadzal je to vhodné. Nasledovný

JavaScript rieši problém tlačidla späť.

```
var sessionFrame = document.getElementById("sessionFrame");
var doc = sessionFrame.contentDocument;
if (doc == undefined) { // Internet Explorer
      doc = sessionFrame.contentWindow.document;
}
doc.open();
doc.write(someNewState);
doc.close();
```

Pri každom zápise je v histórii prehliadača vytvorený záznam. Zmena musí byť robená vrátane funkcií open() a close().

Druhý spôsob je založený na tom že pomocou JavaScriptu je možné meniť časť URL za znakom '#'. Pri vyvolaní akcie ktorá má byť zvrátená tlačidlom „späť“ je preto potrebné zmeniť URL za týmto znakom a zaznamenať čo je potrebné spraviť aby sa aplikácia dostala do stavu pred touto akciou. Na jednoduché implementovanie tohto postupu existuje veľmi jednoduchý framework (26).

5.3 Pridanie medzi obľúbené a posielanie linkov

Pri pridaní linku medzi obľúbené sa ukladá len samotný link. Žiadna iná informácia ukladaná nieje. Tak ak má aplikácia v cookies, Flashi, alebo kdekoľvek inde uložené dáta potrebné na to by sa dokázala do stať do stavu, v ktorom momentálne je a užívateľ s ňou pracuje, pracuje aj s týmito dátami, ktoré pri jeho ďalšej aktivite môžu byť zmenené. To do značnej miery znemožňuje uloženie si konkrétneho stavu, do ktorého by sa dalo neskôr vrátiť.

Úplne iná situácia je ak sa všetky pracovné premenné priebežne ukladajú do URL. Kvôli optimalizácii by šlo použiť aj nejaký druh kompresie a do približne 2000 znakov, ktoré ostanú po odpočítaní znakov pred '#', sa dá dostať pomerne veľa informácií.

Samozrejme, že nemusia byť v tomto priestore uložené hodnoty premenných, môže

ísť o akúkoľvek informáciu, ktorú dokáže JavaScript správne interpretovať a dostať sa do žiadaného stavu.

5.4 Ak sa niečo vykonáva, alebo čaká na server užívateľ by to mal vedieť

Užívateľ je zvyknutý na isté správanie sa internetových stránok a po kliknutí očakáva že stránka sa mu zmení. Pri AJAXe sa však niekoľko sekúnd nemusí stať nič, dokiaľ nepríde zo serveru odpoveď. Je preto vhodné pridať element ktorý užívateľovi oznámi že jeho požiadavka bola zaznamenaná a pracuje sa na nej.

5.5 Bezpečnosť a dôvernosť

Pri každej požiadavke je nutné na serveri dôkladne kontrolovať či užívateľ vykonáva len takú činnosť ako mu jeho úroveň práv dovoľuje a nespoliehať sa že požiadavka bola vygenerovaná príslušným JavaScriptom v ktorom sú bezpečnostné kontroly implementované. Ani sa spoliehať na to že niektoré typy požiadaviek sa nemôžu nikdy vyskytnúť. Táto kontrola by mala byť dôslednejšia ako pri neAJAXových stránkach pretože požiadavky, ktoré na server prichádzajú sú síce jednoduchšie a jednoznačnejšie, no komunikujú so serverom viacerými vstupmi, a tie sa bránia ťažšie ako jeden.

V aplikáciách kde je dôvernosť údajov dôležitejšia než interaktivita je raritou stretnúť sa s AJAXom. Dôvodom je že komunikácia prebieha po nešifrovanom kanály HTTP a nedokáže komunikovať po šifrovanom HTTPS.

No existuje možnosť ako komunikovať aj cez AJAX dôverne. Existujú JavaScriptové knižnice napr. aSSL(24) ktoré umožňujú šifrovať komunikáciu pomocou RSA šifry.

5.6 Neočakávané správanie a funkcie sa užívateľského rozhrania

AJAX ponúka široké možnosti, ale užívatelia keď sú na internetovej stránke tak od nej čakajú istý druh správania sa. To že sa im stránka po kliknutí na link nezačne celá načítavať, ale načítajú sa len potrebné dáta a tie sa zobrazia uvítajú a budú s tým spokojní.

'Drag&drop' je funkcionalita ktorú by normálny užívateľ v prehliadači neočakával. Je to každopádne zaujímavá a praktická vlastnosť, no musí mať svoje miesto a opodstatnenie.

Podobne aj klikanie pravým tlačidlo myši je funkcionalita očakávaná a používaná v desktopových aplikáciách, no vrámci prehliadača užívateľa nenapadne.

AJAX je nástroj na uľahčenie práce s webaplikáciou, nie zaujímavá hračka pre programátorov. A tak ho treba používať tam kde to má zmysel a takým spôsobom aby to bolo užívateľovi na osoh.

6 Subjektívna analýza súčasného stavu a predstava o budúcom vývoji

Posledná kapitola tejto práce obsahuje výrazne viac osobných subjektívnych názorov autora ako predchádzajúcich 5. Vyjadruje to k čomu dospel počas štúdia problematiky, ktorej sa venoval ako svojej diplomovej práci.

V globalizovanom trhu akým internet je môžu zásadné technologické zmeny uskutočniť len veľkí hráči, ktorí dokážu svoje technológie pretlačiť a dosiahnuť aby užívatelia mali na svojich počítačoch nainštalovaného klienta potrebného pre ich technológiu. Na to aby technológia typu XAML/XUL/Flash... mohla využiť celý potenciál ktorý má je potrebné aby bola dostupná na veľkom množstve počítačov, či iných zariadeniach. Kým nebude bežným vybavením počítača môže to byť zásadná prekážka pre manažérske rozhodnutie vybrať si práve túto platformu pre vývoj aplikácie.

Malým firmám podľa môjho názoru ostáva len mať dobrý nápad na využitie existujúcich možností s ktorým sa im podarí presadiť. Príkladom že to ide je projekt youtube.com, umožňujúci zdieľať a sledovať videá cez internet. Spoločnosť založená vo februári 2005, bola po roku a pol existencie odkúpená spoločnosťou Google za 1,65 miliardy dolárov (48,5 miliardy korún) 9. októbra 2006 (29). S pribúdajúcimi možnosťami sa objavia ďalšie možnosti a priestor pre dobré nové zaujímavé nápady nápady.

6.1 Microsoft

Pri písaní tejto práce ma napadla nasledovná myšlienka: dôvod prečo je vývoj, najpoužívanejšieho internetového prehliadača súčasnosti tak strašne pozadu oproti štandardom je ten že Microsoft si uvedomuje že platfomovou nezávislosťou ktorú ponúkajú webové riešenia môže stratiť postavenie na trhu, ktoré sa mu podarilo vybudovať aj tým že užívatelia, ktorí ho raz začali používať s tým nemôžu prestať bez toho aby nemuseli riešiť

množstvo problémov ktoré im vznikne prechodom na inú platofrmu a následným prechodom na nový softvér.

Aj CEO spoločnosti Microsoft povedal že s vydaním nového internetového prehliadača čakali príliš dlho a prisľúbil pravidelnejšie aktualizácie (30), tohto možno 2. najpoužívanejšieho softvéru na svete, otázkou je či to nebol a stále nieje zámer.

Microsoft sa vždy vyznačoval veľmi dobrou marketingovou stratégiu a v roku 2006 sa výrazne viac venoval grafickému subsytému nového Windows Vista ako implementácii štandardu CSS2 do Internet Explorera, (štandard CSS2 je pritom z roku 1998 (15), teda v dnešnej dobe má už 9 rokov). Dalo by sa to interpretovať aj ako snaha o robenie obštrukcií vývojárom web aplikácií, ktorých používanie môže zjednodušiť užívateľom prechod na iný operačný systém. Keby sa webaplikácie robili jednoduchšie bolo by ich viac a užívateľ by nebol natoľko viazaný na svoj operačný systém a aplikácie v ňom nainštalované. Veľmi ťažko sa ako webová aplikácia bude presadzovať taká ktorú v plnej miere nepodporuje najrozšírenejší prehliadač.

6.2 Google

Google-u sa podarilo z vyhľadávača sa vypracovať na službu, ktorej možnosti využívajú stovky miliónov užívateľov. Nebol prvý kto využíval AJAX, no bol to on kto sa zaslúžil o to že sa AJAX dostal do povedomia. A priniesol webaplikácie z podnikovej sféry obyčajným ľuďom.

Problémom aplikácií ktoré ponúka Google, je že koncentrujú veľké množstvo osobných údajov. Uvažujme aktuálne aplikácie súvisiace s osobnými údajmi, ktoré ponúka: Gmail, Blog, Calendar, Picasa, Talk, Pagecreator. Spreadsheet a netreba zabúdať na že Google vie aj to čo vyhľadávame. A v neposlednom rade Google je exepert na vyhľadávanie vo veľkom množstve dát.

Takáto silná koncentrácia osobných údajov sa môže stať bezpečnostným rizikom,

ktoré si ľudia časom môžu uvedomiť. A môže im začať prekážať, že ich majú koncentrované na jednom mieste. Firmy tieto aplikácie pravdepodobne nikdy používať nebudú, ale pre obyčajných užívateľov služby, ktoré Google ponúka zadarmo sú veľmi lákavé.

6.3 Pohľad na budúcnosť (web)aplikácií

V blogu Paula Grahama,(8) ktorý o sebe tvrdí že vytvoril prvú webaplikáciu som našiel zaujímavý článok, ktorý vyjadruje môj názor odkiaľ a kam smeruje vývoj aplikácií ako takých, nielen webaplikácií.

Microsoft vládol počítačovému svetu asi 20 rokov, potom ako túto pozíciu zobral spoločnosti IBM, ktorá mu vládla od jeho vzniku. Momentálne sa v tomto svete stále výraznejšie presadzuje spoločnosť Google, ktorá sa vypracovala na svetového lídra v oblasti inovácií na internete. To čo sa nepodarilo spoločnosti Yahoo, ktorá akoby sa zľakla Microsoftu a svoje miesto na trhu si našla ako „mediálna spoločnosť“ namiesto toho aby využila možnosti ktoré sa jej poskytovali ako technologickej spoločnosti.

Google po tom ako sa stal najpoužívanejším vyhľadávačom ukázal že vie oveľa viac. V roku 2005 keď na svet priviedol Gmail. Týmto produktom ukázal ako veľa sa dá spraviť s internetovo orientovaným softvérovými technológiami ktoré neskôr dostali názov AJAX.

Iróniou je že práve Microsof bol ten kto dal AJAXu základný kameň, tým bol objekt XMLHttpRequest, pomocou ktorého prehliadač komunikuje so serverom a získava tak potrebné informácie. Pôvodne sa nové informácie dali získať len načítaním celej stránky. Keď Microsoft tento objekt potreboval a implementoval pre svoj Outlook Webaccess, ťažko povedať či si uvedomoval kam to až môže dospieť a že práve vďaka tomuto si bude môcť pomocou ich prehliadača vytvoriť ktokoľvek vlastnú web aplikáciu ktorá bude bude platformovo nezávislá, bude spájať výhody desktopových a klient-server aplikácií a v konečnom dôsledku môže ukončiť nadvládu Microsoftu nad svetom počítačov.

Postupne nájdeme na webaplikácie úplne všetko už dnes existujú ekvivalenty k Micorosoft Office. A aj editovanie obrázkov v štýle Photoshopu časom určite príde (na úvod tu máme http://snipshot.com/ nemá síce príliš veľa funkcií, ale ako základ to vyzerá sľubne). So zrýchľujúcim sa internetovým pripojením prestane byť používanie desktopu a priamo v ňom nainštalovaných programov nutnosťou a ľudia si zvyknú pracovať na serveri na ktorý sa dá pripojiť odkiaľkoľvek a pokračovať v práci začatej na inom počítači z úplne iného miesta.

Konečne aj na slovensku sa začína rozvíjať skutočne vysokorýchlostné pripojenie k internetu technológiou FTTH, ktorá umožňuje rýchlosti o ktorých sme pred pár rokmi ani neuvažovali že by boli reálne použiteľné v domácnosti. A čím väčšia je rýchlosť spojenia tým viac aplikácií môže byť umiestnených na serveri.

Všetky nové významné udalosti okolo počítačov sa v posledných rokoch konajú na internete. Ani príchod Windows Vista, nebol zatiaľ taký výrazný ako youtube.com. No možno nás nový Windows ešte prekvapí. Potenciál skrytý za skratkou XAML je vskutku zaujímavý.

Po vzhliadnutí prezentačného videa technológie Silverlight, prvých praktických využití tejto technológie ako aj aplikácií využívajúcich všetky možnosti ponúkané XAML som si uvedomil, že záver tejto práce musím prepracovať.

Pôvodne som chcel písať o tom ako sa stmieva nad Microsoftom a ako môžu webaplikácie zapôsobiť na trhovú dominancia tejto firmy, a ako sa táto dominancia v dôsledku ich masívneho využívania môže skončiť. No zdá sa, že v Microsofte dospeli k tomu istému záveru a okrem toho našli aj riešenie ako si svoje postavenie udržať.

Ak vychádzame z predpokladu, že webaplikácie sú smer kam smeruje vývoj. A tieto aplikácie budeme môcť používať z ľubovolného zariadenia, nielen počítača či notebooku, ale i z PDA, alebo mobilu. Tak prirodzeným krokom Microsoftu je spraviť najlepší prehliadač takýchto aplikácií. Najlepšie taký aby fungoval len na Windowse.

A realita? Najlepší prehrávač webových aplikácií je skutočne momentálne Windows Vista, ktorý ako jediný dokáže využiť všetky ponúkané možnosti XAML. Pre ostatných užívateľov tu je Silverlight, ktorý však v Microsofte nespravili pre Linux, svojho najväčšie konkurenta.

Každopádne webaplikácie ktoré sa na tejto platforme v najbližších rokoch vytvoria, podľa môjho názoru budú za hranicami toho čo dnes bežne používame.

7 Záver

V posledných rokoch sme svedkami prudkého rozvoja webaplikácií, ako aj technológií na ich tvorbu. Užívatelia tak majú možnosť pracovať s interaktívnym užívateľským rozhraním z pohodlia internetového prehliadača. To im umožňuje nech sú kdekoľvek na svete mať k dispozícii svoju aplikáciu so svojimi dátami bez toho aby boli viazaný na vlastný počítač či notebook. Popularita takýchto aplikácií bude v najbližších rokoch naďalej narastať a v blízkej dobe sa podla mňa dočkáme toho, že akákoľvek aplikácia bude mať svoj ekvivalent medzi webapikáciami na porovnateľnej úrovni ako je dnešná desktopová verzia.

Cieľom práce bolo zhrnúť a porovnať najčastejšie používané technológie, ktorými sú AJAX, Javu a Flash s tými ktoré sa ešte nestihli presadiť, XUL a XAML, no majú svoje prednosti, ktoré im otvárajú možnosti vo svete aplikácií, ktoré zmenia náš pohľad na počítač a programy v ňom nainštalované. Zdôrazniť pozitívne vlastnosti, ale i obmedzenia jednotlivých technológií z viacerých uhlov pohľadu.

Čitateľ, ktorý má v pláne vytvoriť vlastnú webaplikáciu a stojí pred rozhodnutím ktorú technológiu si vybrať, by po prečítaní tejto práce mal by mať jasno v tom ktorá technológia je pre jeho potreby najvhodnejšia a prečo. Ako ja podrobne rozobraté percentuálne zastúpenie užívateľov, ktorí jednotlivé technológie nebudú schopní z rôznych dôvodov jeho aplikáciu schopní využívať a prečo.

Ďalej sa v práci podrobnejšie venujem momentálne najrozšírenejšej technológii na tvorbu interaktívnych webových aplikácií – AJAXu. A rozoberám možnosti riešenia najčastejších problémov, s ktorými sa pri tvorbe aplikácií využívajúcich AJAX programátori stretávajú.

Použitá literatúra

(1) Silverlight homepage <http://silverlight.net/>

(2) Wikipedia AJAX (programming) <http://en.wikipedia.org/wiki/AJAX>

(3) Alex Bosworth's Weblog: Ajax Mistakes
<http://alexbosworth.backpackit.com/pub/67688>

(4) Jesse James Garrett: Ajax: A New Approach to Web Applications
<http://www.adaptivepath.com/publications/essays/archives/000385.php>

(5) eyeOS homepage <http://eyeos.org/>

(6) Jonathan Boutelle: Flash: what is it good for?
<http://www.jonathanboutelle.com/mt/archives/2005/11/flash_what_is_i.html>

(7) Macromedia Flash and Shockwave Players
<http://www.adobe.com/products/player_census/npd/>

(8) Paul Graham: Microsoft is Dead
<http://www.paulgraham.com/microsoft.html>

(9) Adobe Flash Player Browser Support Matrix <http://www.adobe.com/go/tn_14159>

(10) ECIS Media Release: With Vista Microsoft continues its illegal practices
<http://www.e-c-i-s.org/news/2007_jan26.htm>

(11) Can XAML be used instead of HTML?
<http://www.kudzuworld.com/blogs/Tech/20061231.no.aspx>

(12) Mozilla developer center: The Joy of XUL
<http://developer.mozilla.org/en/docs/The_Joy_of_XUL>

(13) XUL Planet: Why use XUL? <http://www.xulplanet.com/tutorials/whyxul.html>

(14) Microsoft: XAML Case Study

<http://www.microsoft.com/casestudies/casestudy.aspx?casestudyid=201270>

(15) Cascading Style Sheets, level 2 CSS2 Specification <http://www.w3.org/TR/REC-CSS2/>

(16) WHATWG: Web Applications 1.0 <http://www.whatwg.org/specs/web-apps/current-work/>

(17) Secutiryspace: Technology Penetration Report <http://www.securityspace.com/s_survey/data/man.200703/techpen.html>

(18) W3C: Roadmap for Accessible Rich Internet Applications (WAI-ARIA Roadmap) <http://www.w3.org/TR/aria-roadmap/>

(19) FOLDOC: state <http://foldoc.org/index.cgi?query=state&action=Search>

(20) sitepopint: State of Web Development 2006 <http://sitepoint.com/report2006/claim/dd>

(21) Read/WriteWeb: The State Of Web Development - Ajax set to surpass Flash in '07 <http://www.readwriteweb.com/archives/the_state_of_web_development.php>

(22) Mark Finkle: XUL/E - What If <http://starkravingfinkle.org/blog/2006/12/xule-what-if/>

(23) [Editorial Draft] State in Web application design <http://www.w3.org/2001/tag/doc/state.html>

(24) aSSL: RSA for AJAX <http://assl.sullof.com/assl/>

(25) w3schools: Web Statistics and Trends <http://www.w3schools.com/browsers/browsers_stats.asp>

(26) Really Simple History <http://codinginparadise.org/projects/dhtml_history/README.html>

(27) Adobe: Flash support center <http://www.adobe.com/support/flash/action_scripts/local_shared_object/>

(28) Microsoft: Maximum URL length is 2,083 characters in Internet Explorer

<http://support.microsoft.com/kb/208427>

(29) Google: Google To Acquire YouTube for $1.65 Billion in Stock <http://www.google.com/press/pressrel/google_youtube.html>

(30) Zive: Bill Gates o IE7: Čakali sme príliš dlho <http://zive.sk/h/Bleskovky/AR.asp?ARI=116789>

(31) Adobe: Adobe Flash Player Version Penetration <http://www.adobe.com/products/player_census/flashplayer/version_penetration.html>

(32) Jim Hugunin: Thinking Dynamic <http://blogs.msdn.com/hugunin/archive/2007/04/30/a-dynamic-language-runtime-dlr.aspx>

(33) Mozilla: XUL:IDE <http://wiki.mozilla.org/XUL:IDE>

(34) Swik: Places To Use Ajax <http://swik.net/Ajax/Places+To+Use+Ajax>

(35) AnfyFlash: Why use Flash applets instead of Java applets <http://www.anfyflash.com/flashinsteadofjava.html>

(36) Flash or Java. Which is better? <http://blog.pullur.com/2007/04/06/flash-or-java-which-is-better/>

(37) mozilla: XUL accessibility guidelines <http://developer.mozilla.org/en/docs/XUL_accessibility_guidelines>

(38) MSDN: XAML Accessibility Best Practices <http://msdn2.microsoft.com/en-gb/library/aa350483.aspx>

(39) Introduction to Presenting Scientific and Medical Data on the Web using Scalable Vector Graphics <http://www.medicalcomputing.net/svg_tutorial.html>

(40) SVG: Firefox 1.5 <http://www.svg.org/story/2005/11/30/44555/315>

(41) SVG: Shipping and Announced SVG Phones <http://www.svg.org/special/svg_phones>

(42) Adobe: Scalable Vector Graphics <http://www.adobe.com/svg/overview.html>

(43) PHPTR: Introduction to SVG <http://www.phptr.com/articles/article.asp?p=99036&seqNum=2&rl=1>

(44) Wikipedia: Scalable Vector Graphics <http://en.wikipedia.org/wiki/Scalable_Vector_Graphics>

(45) SVG: Create SVG Themes with Sony Ericsson Themes Creator <http://www.svg.org/story/2005/10/25/16936/714>

(46) OpenLaszlo homepage <http://www.openlaszlo.org/>

(47) AFLAX homepage <www.aflax.org>

(48) Adobe: Apollo:DeveloperFAQ <http://labs.adobe.com/wiki/index.php/Apollo:DeveloperFAQ>

(49) DDJ: RAD XAML with Microsoft Expression Blend <http://www.ddj.com/dept/webservices/198000863>

(50) MSDN: XAML Overview <http://msdn2.microsoft.com/en-us/library/ms752059.aspx>

(51) SCRIPTOL: Which Interface for a Web Application? <http://www.scriptol.com/ajax/ajax-xul-xaml.php>

(52) JAVA: Applets <http://java.sun.com/applets/>

(53) WHATWH: Plans for HTML6<http://blog.whatwg.org/html6-plan>

(54) W3C: Web Content Accessibility Guidelines 1.0 <http://www.w3.org/TR/WCAG10/wai-pageauth.html#tech-scripts>

(55) AjaxPatterns: Frameworks <http://ajaxpatterns.org/Ajax_Frameworks>

(56) NEWS.COM: Silverlight on Linux? We're in, says Mono founder <http://news.com.com/8301-10784_3-9714669-7.html>

(57) Silverlight: Presentation video <http://download.microsoft.com/download/8/5/0/85096922-090d-4dfa-96b6-

f74810411973/FullCut2.wmv>

(58) ADOBE: Flash 8 Accessibility Design Guidelines <http://www.adobe.com/resources/accessibility/flash8/best_practices.html>

(59) IBM: AJAX Accessibility Overview <http://www-03.ibm.com/able/resources/ajaxaccessibility.html>

(60) Miguel de Icaza: Silvelight <http://tirania.org/blog/archive/2007/May-01.html>

(61) Microsoft: Silverlight Developer reference <http://download.microsoft.com/download/f/2/e/f2ecc2ad-c498-4538-8a2c-15eb157c00a7/SL_Map_FinalNET.png>

(62) Adobe: Flash TechNote <http://www.adobe.com/go/tn_14159>