

ČESKÁ ZEMĚDĚLSKÁ UNIVERZITA

PROVOZNĚ EKONOMICKÁ FAKULTA

Katedra informačních technologií
obor Informatika



TÉMA BAKALÁŘSKÉ PRÁCE

Rich Internet Applications

Vedoucí bakalářské práce:
Ing. Jíří Vaněk, Ph.D.

Autor:
Zdeněk Klůc © 2009
ČZU v Praze

Čestné prohlášení

Prohlašuji, že svou bakalářskou práci „Rich Internet Applications“ jsem vypracoval samostatně pod vedením vedoucího bakalářské práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou citovány v práci a uvedeny v seznamu literatury na konci práce. Jako autor uvedené bakalářské práce dále prohlašuji, že jsem v souvislosti s jejím vytvořením neporušil autorská práva třetích osob.

V Praze dne 22. 4. 2009.

.....

Zdeněk Klůc

Poděkování

Rád bych na tomto místě poděkoval Ing. Jiřímu Vaňkovi, Ph.D. za cenné rady a ochotu při vedení mé bakalářské práce. Dále bych rád poděkoval pánům Illyovi Pavlovovi a Ladislavu Košťákovi ze společnosti Oracle Czech s.r.o. za poskytnutí cenných informací.

Rich Internet Applications

Souhrn

Bakalářská práce pojednává o nové generaci webových aplikací označované jako Rich Internet Applications. Práce rovněž charakterizuje jejich obecné rysy a představuje několik současných technologií, které slouží pro jejich vývoj a běh na straně koncového uživatele. Ve vybraných ukázkách zároveň prezentuje konkrétní aplikace v nejrůznějších oblastech za použití uváděných technologií, u kterých autor bakalářské práce provede srovnání dle několika kritérií a nejen na jeho základě se pokusí předpovědět budoucí vývoj této oblasti.

Klíčová slova

Bohaté internetové aplikace (RIA), Ajax, Flash, Flex, AIR, Silverlight, JavaFX Adobe, Macromedia.

Rich Internet Applications

Summary

The thesis speaks about new generation of web applications- commonly called Rich Internet Applications. This work also describes its common features and introduces several nowadays technologies, helping the development of RIA and their operations on the side of end-users. In several cases, this work also present particular applications in different areas with usage of mentioned technologies. Those will be compared based on several criteri, and the reset will help to speak about the near future of the given area.

Keywords

Rich Internet Applications (RIA), Ajax, Flash, Flex, AIR, Silverlight, JavaFX Adobe, Macromedia.

Obsah

1 Úvod.....	3
2 Cíl práce a metodika.....	4
3 Historický vývoj.....	5
3.1 Grafické uživatelské rozhraní.....	5
3.2 Značkovací jazyky.....	6
3.2.1 SGML.....	7
3.2.2 HTML.....	7
3.2.3 CSS.....	8
3.2.4 XML.....	9
3.3 Cesta k dynamickému webu.....	10
3.3.1 CGI.....	10
3.3.2 Aplety.....	10
3.3.3 Servlety, ASP a další.....	11
3.3.4 JavaScript.....	11
3.3.5 Flash.....	12
3.4 Základní problém.....	13
4 Co je Rich Internet Application.....	14
4.1 Teoretický základ.....	15
5 Přehled dostupných technologií.....	18
5.1 Ajax.....	19
5.1.1 Document Object Model.....	19
5.1.2 XMLHttpRequest.....	20
5.1.3 Knihovny a aplikační rámce.....	21
5.1.4 JavaScript mimo web.....	25
5.2 Adobe Flash Platform.....	26
5.2.1 Flash Player.....	27
5.2.2 Flash.....	28
5.2.3 Flex.....	30
5.2.4 OpenLaszlo.....	32
5.2.5 AIR.....	33
5.2.6 Budoucnost v maximální integraci.....	34
5.3 Windows Presentation Foundation.....	34
5.3.1 Silverlight.....	35
5.4 JavaFX.....	37
6 Srovnání vybraných technologií.....	37

7 Implementace RIA – příklad řešení	41
7.1 Google Maps, Mail, Docs... ..	42
7.2 Meebo.com	43
7.3 Photoshop.com	43
7.4 NASDAQ Market Replay	44
7.5 Oracle CRM Gadgets	45
7.6 eBay Desktop	45
7.7 G.ho.st	46
7.8 NBColympics.com	47
7.9 Windowsvista.si	48
8 Očekávaný vývoj.....	49
9 Závěr	50
10 Seznam literatury	51
11 Přílohy	54
12 Seznam obrázků	56
13 Seznam tabulek a grafů	58

1 Úvod

Internet, celosvětový systém počítačových sítí představuje informační prostředek, který v dějinách lidského snažení snad nemá obdoby. Pokud se v souvislosti s rozvojem letecké dopravy mluvilo o zkracování vzdáleností na zemi, příchod Internetu veškeré vzdálenosti smazal, a celkově tak ovlivnil životní styl lidí, pro které se stal součástí každodenní reality. S rozšiřováním jeho nejznámější služby Word Wide Web je proto snaha co nejvíce aktivit přesouvat do tohoto prostředí, jehož současný rozvoj dosahuje takových rozměrů, že je prakticky nemožné jednoduše charakterizovat jeho stav.

Nejvíce se asi vžilo příznačné označení metaforou Web 2.0, jejímž autorem je nakladatel a technologický vizionář Tim O'Reilly. Právě s Webem 2.0 souvisí silný nástup webových aplikací, které dosahují čím dál vyšší úrovně, takže řada z nich se stává potenciální konkurencí pro ty klasické - běžící na desktopu. Mění se tak celkový pohled na software jako na produkt, který se posunuje spíše do oblastí služeb poskytovaných prostřednictvím Internetu. V tomto případě současnou nadějí představuje nová generace takzvaných Rich Internet Application (RIA, neboli bohaté internetové aplikace).

Počáteční slovo Rich klade důraz na jejich bohatost s ohledem na kvalitu uživatelského rozhraní. Zejména vizuální složka a uživatelský komfort dnes představuje nezanedbatelnou součást softwarového vývoje, který z pohledu do nedávné historie byl právě v tomto aspektu silně limitován. Uživatelské rozhraní prvních webových aplikací bylo strohé a plnilo pouze základní funkcionální složku. Nízká přenosová rychlost neumožňovala přenášet objemná grafická data a omezené možnosti tehdejších technologií nemohly poskytnout dostatečné prezentační schopnosti s přihlédnutím na složitou interaktivitu.

Rich Internet Applications by tento stav měly změnit, což představuje možnost překlenout rozdíl mezi aplikacemi webovými a mezi těmi klasickými, které jsou nainstalované na osobním počítači.

2 Cíl práce a metodika

Práce si klade za cíl demonstrovat současné technologie používané k tvorbě Rich Internet Application a následně charakterizovat a ukázat jejich základní principy a jejich praktické využití. Dále si klade za cíl zachytit současné a nastupující trendy v této oblasti a předpovědět tak další vývoj do budoucnosti. S ohledem na téma a rozsah bakalářské práce se autor bude zaměřovat pouze na problematiku klientské části těchto aplikací, protože právě ta tvoří jejich inovaci.

Úvodní krátké kapitoly zahrnuté pod názvem “Historický vývoj“ pojednávají o stručné historii a vývoji technických způsobů prezentace informací na Internetu, které jsou nezbytné pro pochopení následujících vývojových milníků, které vedou k nástupu RIA.

Následující kapitola „Co je Rich Internet Application“ je brána jako logické vývojové vyústění kapitoly předcházející. Snaží se obecně specifikovat danou problematiku a určit základní charakteristiky, které jsou nezaměnitelné pro bohaté internetové aplikace. Bude vysvětlena základní koncepce a filosofie, kterou tyto aplikace splňují.

V nejobsáhlejší části „Přehled dostupných technologií“ budou vysvětleny a probrány současně nejpoužívanější technologie pro tvorbu RIA. Budou rozděleny do dvou velkých skupin z hlediska prostředí, na kterých je umožněno jejich fungování a které jsou tím pádem specifické pro jejich vývoj. Největší pozornost bude věnována platformě Flash, se kterou má autor osobní zkušenost a která dle jeho názoru v oblasti webu, multimédií a grafiky určují a zažívá v poslední době největší progresy. Poté se autor pokusí vybrané technologie srovnat dle několika kritérií.

Vzhledem k tomu, že už dnes je na Internetu k nalezení mnoho příkladu nasazení RIA, proto bude další kapitola ukázkou konkrétních realizovaných implementací napříč celým spektrem možného využití.

Poslední část se bude snažit nastínit budoucí vývoj ohledně perspektivního rozvoje těchto webových aplikací obecně. Autor tak učiní na základě převzatých názorů odborníků v dané oblasti, ale i dle názorů vlastních.

3 Historický vývoj

Dnes Internet představuje nejrychleji se rozvíjející informační technologii, která by letos oslavila již třicet let a s jeho službou World Wide Web by to bylo let dvacet. Ve vztahu k vývoji lidské společnosti by se dalo namítat, že těchto několik desítek let znamená pouhý okamžik nebo krátkou epizodu, avšak z pohledu výpočetní techniky se jedná již o poměrně dlouhou historii. Rich Internet Applications patří k dnešním vrcholům, které tuto službu využívají a nelze předpokládat, že se objevily jako zcela nový koncept bez jakékoliv předešlé evoluce. Proto tato bakalářská práce věnuje i svou část historii webových technologií, které RIA předcházely nebo na kterých dosud staví své základy.

Kapitola Historický vývoj je zaměřena na vývoj technologií, které posouvaly možnosti World Wide Webu stále více kupředu. Proto možná nelogicky jako první následuje text, který tak úplně s vývojem webu nesouvisí. Autor však považuje za nutné toto téma alespoň rámcově rozebrat, protože se bude objevovat v průběhu celé bakalářské práce.

3.1 Grafické uživatelské rozhraní

Grafické uživatelské rozhraní, graphic user interface (GUI), user interface (UI), či prezentační vrstva aplikace jsou všechno synonyma, pro jediný aspekt, který počítače provází tak dlouho, že si jeho vliv možná mnoho lidí neuvědomuje a bere ho jako úplnou samozřejmost.

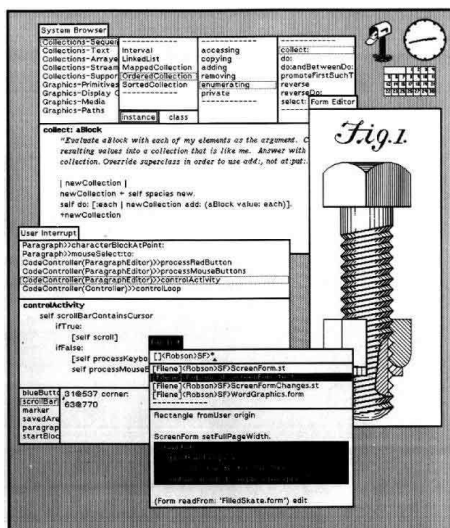
V souvislosti s historií výpočetní techniky, vždy byla potřeba nějakým způsobem zprostředkovat výměnu informací mezi člověkem a strojem. Přes prvotní zobrazování dat prostřednictvím tištěných médií nebo děrných štítků se revoluční komponentou výstupního zařízení stala obrazovka, na které bylo možné zobrazovat data v textové formě.

V tomto ohledu se důležitým obdobím stala 70. léta, kde ve výzkumných laboratořích společnosti Xerox se začalo pracovat na vývoji nového ovládání počítače. Vnikl tak první počítač Xerox Alto, který na své obrazovce prezentoval informace formou GUI, tak jak je známe dnes. Vyjádřenou hodnotou tohoto vynálezu snad jen může být analogie kniha versus časopis, kde uživatel odkázaný jen textovou formou výstupu mohl teď využít grafické zkratky, která mu zprostředkovala daleko přístupnější pohodlnější práci s počítačem.

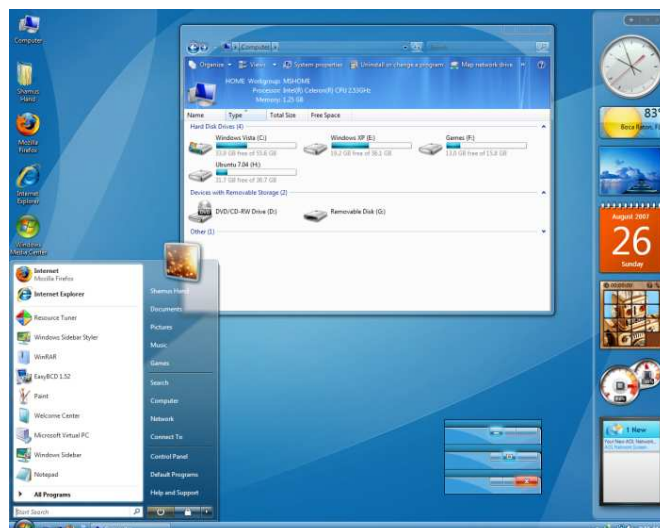
V letech osmdesátých využily příležitosti společnosti jako Apple a Microsoft, a vybavily GUI své operační systémy a zasloužily se tak o masové rozšíření. Už jen samotný název OS Windows si dodnes nese své jméno díky svým grafickým oknům.

Graphic user interface se stal průmyslovým standardem v softwarovém inženýrství a přizval tak do oblasti, kde především vynikali programátoři, novou profesní skupinu, jenž zapříčinila vznik nových oborů zabývajících se interface designem, ergonomií uživatelského rozhraní, a posléze webdesignem. V současnosti existují celé firmy, které se na danou problematiku specializují. Příkladem může být společnost The Skin Factory stojící za vizuální podobou vypalovacího programu Nero nebo návrháři z Icon Factory, kteří připravili sadu ikon pro operační systém Windows Vista. [9] Softwarový vývoj se tak musel začít orientovat i na grafickou stránku věci, která se stala jeho nemalou marketingovou hodnotou.

Více informací v [7]



Obrázek 1. Operační systém počítače Alto(1973)



Obrázek 2. OS Windows Vista (2007)

Grafické provedení je výrazně kvalitnější, avšak základní koncepce zůstává bezmála 35 let stejná. Za povšimnutí stojí prvek ručičkových hodin v horním pravém rohu...

3.2 Značkovací jazyky

V obecnějším pohledu se synonymem pro web stal značkovací jazyk HTML a doposud ho lze považovat za jeden z jeho elementárních stavebních prvků. Značkovacích jazyků vznikla celá řada, ale vzhledem k tématu této práce je nejdůležitější právě HTML a jeho předchůdce SGML. Oba dva se staly základem pro své aktuální současníky, kteří z nich více či méně doposud vycházejí.

3.2.1 SGML

V 80. letech ze strany společnosti IBM a dalších velkých průmyslových firem vznikla potřeba standardu pro výměnu informací mezi různými počítači. Roku 1986 se stal výsledkem jejich společné dohody SGML (Standard Generalized Markup Language) standardizovaný obecný značkovací jazyk. Tvořil součást projektu ODA (Open Document Architecture), který si kladl za cíl poskytovat standardní prostředky pro předávání a zpracování dokumentů v elektronické podobě ve formátu nezávislém na softwarové a hardwarové platformě v rámci jednotlivých informačních systémů.

SGML představuje textový dokument, jehož jednotlivé prvky jsou oddělené značkami. Význam těchto značek je zaznamenán ve slovníku DTD (Document Type Definition), který není jejich počtem nijak omezený a umožňuje definovat sady vlastních prvků. V tomto ohledu je SGML chápán jako metajazyk, na jehož základě mohou být vyvářeny další podmnožiny vlastních jazyků. Nadneseně tak lze říci, že je schopné nadefinovat snad všechno počínaje matematickými výrazy, přes abkudu konče aztéckým písmem. Interpreti jednotlivých dokumentů tak využívají slovníku DTD, díky kterému jsou schopny určit jeho strukturu pro požadovaný výstup. SGML se tak zajímá pouze o obsah a nikoliv o formu prezentace dat. Ta je určena až na základě konkrétní situace a instrukcí daného interpreta, který může představovat laserová tiskárna, textový editoru nebo webový server.

3.2.2 HTML

Za počátkem služby World Wide Web paradoxně nestojí žádné velké softwarové firmy, ale místo jejího vzniku vede do švýcarského CERNu. Toto výzkumné středisko zaměstnává mnoho vědců, kteří potřebují na síti vzájemně sdílet nejrůznější textové dokumenty spojené s jejich vědeckou činností. Pro tyto požadavky bylo jednoduchou volbou využití nějakého známého značkovacího jazyka. Tehdy poměrně nové SGML díky své velké komplexnosti vyžadovalo vysoké nároky na výpočetní výkon, takže k webovým účelům nebylo vhodné.

Na tomto základě roku 1990 vzniklo HTML (*Hypertext Markup Language*), jehož autorem byl tehdejší pracovník výpočetního střediska CERN Tim Berners-Lee.

Při jeho tvorbě vycházel z jazyka SGML, který se výrazně redukoval pro větší jednoduchost kódu. Berners-Lee definoval značkovací jazyk pro prezentaci textu, který obohatil o takzvaný hypertext, tedy o funkci odkazovat se z jiného dokumentu na druhý

v rámci celé sítě. Současně vytvořil první webový server, nový komunikační model prostřednictvím protokolu HTTP (HyperText Transfer Protokol) a první webový prohlížeč, který nazval WorldWideWeb.

Jazyk HTML je charakterizován množinou značek (tagů) a jejich atributů, což jsou parametry vyjadřující jejich jednotlivé vlastnosti. Mezi tagy se uzavírají části textu, které tak určují jeho význam, tyto uzavřené části se nazývají elementy. Popis pravidel pro jednotlivé značky a atributy je definován v již zmiňovaném slovníku DTD, který říká jednotlivým webovým prohlížečům jak má HTML dokument konkrétně zpracovat (parsovat).

Díky své jednoduchosti se velice rychle rozšířilo a v 90. letech se začaly objevovat první komerční prohlížeče, které v konkurenčním boji vylepšovaly HTML navrhováním vlastních tagů, což se bohužel negativně projevilo na kompatibilitě tohoto jazyka mezi jednotlivými výrobci. Na základě této situace vzniklo v roce 1994 konsorcium W3C(World Wide Web Consortium), které se dodnes stará o definování a prosazování webových standardů.

Největšími rivaly v tomto ohledu byl Netscape Navigator a Microsoft se svým Internet Explorer, který ho, na rozdíl od Netscapu, distribuoval se svým operačním systémem zdarma. Jejich boj skončil až u soudu za porušení antimonopolního zákona, přesto Internet Explorer je dnes nejpoužívanějším prohlížečem vůbec.

3.2.3 CSS

CSS(Cascading Style Sheets) neboli tabulky kaskádových stylů se objevily v roce 1997 se čtvrtou verzí HTML. Jedná se o jazyk navržený organizací W3C pro definici zobrazování dokumentů, které jsou napsané pomocí značkovacích jazyků.

Hlavním smyslem CSS je oddělit vzhled stránky od její struktury a obsahu, což měl původně umožnit už jazyk HTML, ale v díky konkurenčnímu boji prohlížečů se vyvinul jiným směrem.

Oddělení vzhledu stránky od obsahu v praxi znamená, že si návrháři definují vlastnosti jednotlivých elementů dokumentu, podobně jako to dělají běžní uživatelé pomocí stylů v běžném textovém editoru. Prostřednictvím CSS se tak mohou globálně měnit jednotlivé vizuální vlastnosti webových stránek.

Kaskádové styly svým příchodem rapidně oživily web, a v současnosti představují nenahraditelného pomocníka pro mnoho profesionálních webdesingerů.

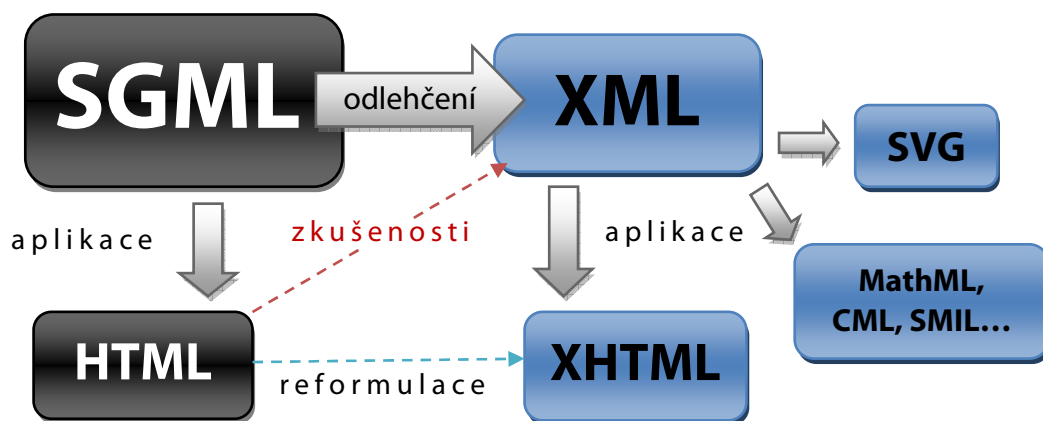
3.2.4 XML

Dalším vývojovým stupněm v oblasti značkovacích jazyků je XML (eXtensible Markup Language), za kterým opět stojí konsorcium W3C. XML je zjednodušenou formou jazyka SGML, takže má jistou příbuznost s HTML, avšak na rozdíl od něho není určen pro vizuální formátování dat, ale pro definici jejich struktury a vnitřních vztahů.

Stejně tak jako u jeho předchůdce se jedná o metajazyk, který umožňuje vytvářet vlastní jazyky pro specifické aplikace, respektive uchovává si možnost definovat vlastní DTD. Na tomto základě vzniklo mnoho derivátů XML, které jsou především určeny pro výměnu dat mezi aplikacemi a pro publikování dokumentů. Pro orientaci je několik z nich uvedeno v následujících řádcích.

- CML (Chemical Markup Language) umožňuje vyjádřit molekulární vztahy v chemii.
- MathML (Mathematical Markup Language) slouží pro charakteristiku matematických symbolů a vzorců.
- RSS (Rich Site Summary), formát pro čtení obsahu z webových stránek.
- SMIL (Synchronized Multimedia Integration Language), koordinuje zobrazování medií na webu.
- XHTML (eXtensible HyperText Markup Language) vzniklo jako vhodná aplikace XML na základě zkušeností s HTML s ohledem na výhody, které tento metajazyk přináší jako je možnost vyjádřit logické uspořádání dat nebo celkově přísnější syntaxe.
- SVG (Scalable Vector Graphics) definuje vektorovou grafiku, která pro svou nízkou datovou náročnost je mnohem vhodnější pro publikování na webu, než bitmapová.

Více informací v [6] a [20]



Obrázek 3. Vztahy mezi značkovacími jazyky.

3.3 Cesta k dynamickému webu

V úplných začátcích byly webové stránky statické a nepředstavovaly nic jiného než elektronickou prezentaci textových dokumentů, což v raných fázích Internetu, kdy ho používaly hlavně vědci pro výměnu informací, nepředstavovalo žádný problém. Avšak s rostoucím počtem uživatelů vstupuje do této oblasti i komerční sféra, a od webových stránek začíná být vyžadováno více než předtím. Mimo jednoduché HTML formuláře, začínají tvůrci jejich obsah obohacovat o skripty, které jim umožní jistou interakci s uživatelem. Internet se tak od své statické podoby přesouvá do další fáze dynamického webu.

3.3.1 CGI

Prvním řešením jak udělat web více dynamický byly CGI (Common Gateway Interface), které jsou na Internetu přítomné již od začátku 90. let. CGI jsou skripty spustitelné na straně serveru, které lze psát v celé řadě jazyků přes Perl po Visual Basic. Na základě požadavků uživatele zajišťují elementární funkce jako generování HTML dokumentů, přístup k databázím a další řadě externích programů.

Na druhou stranu představují určité bezpečnostní riziko, protože dovolují spouštět programy na cizím systému, což v rukou škodlivých uživatelů je snadno zneužitelné. Navzdory svým nedostatkům se používají v určité míře dodnes, ale ve většině případů byly nahrazeny modernějšími skriptovacími jazyky.

3.3.2 Aplety

V roce 1995 společnost Sun Microsystems představila nový objektový programovací jazyk Java, který byl prvotně určen pro ovládání domácích spotřebičů jako ledničky, mikrovlnné trouby nebo set-top-boxy. Shodou okolností Java našla své uplatnění na Internetu díky prohlížeči Netscape Navigator, který jí zařadil do své podpory. Nastala nová éra dynamických stránek - éra apletů.

Apletem je myšlena malá aplikace, napsaná v Javě, která je zabudována do webové stránky. V případě, že klientský prohlížeč podporuje tento jazyk může být aplet interpretován pomocí virtuálního počítače Java Virtual Machine. Přestože mohou aplety provádět na klientské straně mnoho věcí, mají spoustu omezení. Nesmí číst ani zapisovat do souborového systému, nesmí spouštět programy a nesmí provádět další potenciálně nebezpečné operace. Proto jsou spouštěny v ochranné bariéře, obecně nazývané Sandbox. Aplety se staly první opravdovým způsobem jaký by na klientské straně mohly být vyvářeny aplikace. Přestože se poměrně hojně používaly, nikdy se

nestaly silnou technologií. Zejména díky tomu, že pro jejich běh byla vždy vyžadována správná verze Javy, což zamezovalo širšímu rozšíření aplikace. Často se i objevovaly špatně napsané aplikace spojené s vizuálně neatraktivním GUI. Tyto faktory odradily spoustu budoucích uživatelů a vybudovaly tak Javě špatnou popularitu takovým způsobem, že jejich používání při vývoji webových aplikací bylo rovnou považováno za chybu.

3.3.3 Servlety, ASP a další

Navzdory nepovedenému nasazení apletů se Java uchytila na straně serveru pro psaní takzvaných servletů. Servlety představovaly jakousi alternativu k CGI a odbourávaly výše zmiňované problémy s verzováním Javy.

Postupem času servlety byly nahrazeny modernějšími jazyky JSP (Java Server Pages), ASP (Active Server Pages, řešení od Microsoftu), které umožňují oddělení prezentační vrstvy od aplikační logiky. Osvobodily tak programátory od návrhu uživatelského rozhraní, na který se teď mohli soustředit designeři. Objevila se i další řešení jako PHP (Personal Home Page) nebo ColdFusion, ale nakonec všechny řeší stejnou úlohu, dynamicky generovat HTML na straně serveru. To řeší problém s distribucí klientského softwaru, avšak stěží lze porovnávat prezentační možnosti čistého HTML s možnostmi klasických klientských aplikací.

Více informací v [5]

3.3.4 JavaScript

Přibližně ve stejnou dobu představil Netscape společně se Sun jazyk JavaScript. Slovo Java je však součástí jeho názvu pouze z marketingových důvodů a s programovacím jazykem Java jej vedle názvu spojuje jen podobná syntaxe. JavaScript byl vytvořen pro potřeby webdesignéru a vývojářů, kteří neovládali Javu. Pracuje na klientské straně a zapisuje se přímo do HTML kódu nebo může být uložen samostatně v externím souboru, na který se poté dokument odkazuje. JavaScriptem jsou obvykle ovládány různé interaktivní prvky GUI (tlačítka, textová políčka) nebo tvořeny animace a efekty obrázků. Za zmínku stojí, že alternativu k tomuto jazyku si vytvořil i Microsoft s názvem VBScript. Pro svou jednoduchost se časem rozšířil a stal se mocným nástrojem pro vytváření dynamických stránek.

V souvislosti s JavaScriptem vznikl DOM (Document Object Model). Způsob, při kterém prohlížeč interpretuje HTML kód jako množinu objektů, které je možné modifikovat skriptovacím jazykem. Zpočátku byly obě technologie úzce propojeny, ale postupem času se rozdělily jako různé koncepty. DOM se ujala organizace W3C

a JavaScript standardizovala asociace ECMA (European Computer Manufacturers Association) pod názvem ECMAScript.

Příslibem těchto standardů by mělo být stejné chování stránek bez ohledu na to jaký uživatel používá prohlížeč. Avšak v praxi podpora prohlížečů není zcela ideální, což způsobuje, že se stejný skript může chovat různě.

Bohužel díky až někdy v přehnaném designérském využití, různým bezpečnostním problémům a právě zmiňovanému rozdílnému chování stejného kódu se tento jazyk stával terčem velké kritiky. Proto někteří uživatelé neváhali využít možnosti prohlížečů JavaScript vypnout, a mnoho vývojářů se spokojilo bez něj, když při odlaďování kódu komplexních aplikací zažili naprosté psychické vyčerpání.

3.3.5 Flash

S rozvojem interaktivního webu, neodmyslitelně souvisí San Franciská společnost Macromedia. Od roku 1992, kdy byla založena, představila pro tvůrce webových stránek celou řadu nástrojů pro snadnou a efektivní práci na jejich projektech, které se de facto staly standardními nástroji používanými dodnes. V oblasti dynamiky na internetu je především nejzajímavější produktem Macromedia Flash. Jeho první verze vyšla v létě 1996 ještě jako FutureSplash Animator od společnosti FutureWave, která byla Macromedií záhy odkoupena.

Obecně lze říci, že Flash je grafický a animační nástroj, který je určen pro tvorbu interaktivních prvků, animací, prezentací nebo her. Jeho výstupní soubor s hotovou animací je možné vnořit do HTML dokumentu, kde plní interaktivní složku webové prezentace.

Stejně jako další podobná řešení i Flash pro svou interpretaci vyžaduje software na straně klienta pojmenovaný jednoduše Flash Player.

K velkému rozšíření Flashe mezi webovými technologiemi pomohla jeho jednoduchost, kdy tvůrce nemusel mít zkušenosti s programováním na rozdíl od JavaScriptu nebo Apletů. Také díky nízké velikosti výstupních souborů nahradil v oblasti animované grafiky tehdy používaný formát GIF, který se svojí 8 bitovou barevnou hloubkou, triviálním způsobem animace a velkou velikostí nedostačoval a brzdil rozvoj v oblasti bohaté webové grafiky. Flash se stal populárním hlavně mezi grafickými designéry a animátory. Postupem času byl obohacen o jednoduchý objektový programovací jazyk, velice podobný JavaScriptu, jazyk ActionScript, jehož prostřednictvím lze ovládat animace nebo řídit jednotlivé události vytvořené webové aplikace.

Flash se stal synonymem pro interaktivní bohatý webový obsah. Avšak tehdejší malá podpora Flash Playeru v prohlížečích, u rozsáhlejších projektů poměrně vysoká datová velikost výsledných souborů, včetně s nedůvěrou programátorů jako v nástroj, ho zbrzdila v jeho rozšíření mezi standardy internetových aplikací. Skutečná éra Flashe jako silné platformy přišla až o několik let později.

Podrobněji zpracováno v [5]

3.4 Základní problém

S příchodem webových aplikací, se pro mnoho uživatelů stal Internet jako dobrá možnost volby. Na rozdíl od svých protějšků na stolních počítačích, Internetové aplikace nevyžadovaly dlouhosáhlou instalaci, byly nezávislé na operačním systému, uživatel se nestaral o aktualizaci, vždy měl k dispozici nejnovější verzi a mohl s ní pracovat na všech počítačích, kde bylo připojení k síti.

Nicméně přes všechny tyto výhody, webové aplikace trpí zásadními nedostatky vycházející ze samotné povahy webu. To z nich mnohdy dělá těžkopádné řešení, na které je nahlíženo jako na něco druhořadého. Dva nejdůležitější problémy, které je omezují jsou tyto:

- Zvyšující se požadavky na uživatelské rozhraní. Vysoké nároky uživatele na kvalitu a komplexnost UI, kterému vyhovují desktopové aplikace, se na web přenášely velice špatně. A to zejména díky nedostatečným prezentačním možnostem HTML a CSS.
- Striktní návaznost na samotný HTTP protokol a jeho model žádost/odpověď, který brzdil rychlou interakci mezi aplikací a uživatelem.

Často uváděným příkladem, který pomáhá lépe pochopit nedokonalost komunikace přes HTTP je modelová situace zákazníka nakupujícího v elektronickém obchodě. [xxx]

Mějme tedy e-shop, který si ve svém webovém prohlížeči otevře nějaký zákazník, samozřejmě za účelem koupě zboží. Nejspíše bude postupovat dle následujícího scénáře. Uživatel je na úvodní stránce e-shopu a do vyhledávání napíše věc, o kterou má zájem, STRÁNKA 1. Pustí hledání a server zpracuje takto daný požadavek a vrátí zpět odpověď, STRÁNKA 2. Pokud zákazník našel, co hledá, přidá svoje zboží do nákupního košíku, server opět zpracuje požadavek a vrací uživateli data zpět, STRÁNKA 3. Dále může jít do volby zaplacení, STRÁNKA 4, nebo může pokračovat v nakupování a opakovat kroky, respektive STRÁNKY 1-3.

Při volbě zaplacení se na serveru provede výpočet daných položek a jejich cen, které se poté zobrazí v konečné ceně, STRÁNKA 5. Dále je třeba vyplnit fakturační

údaje, STRÁNKA 6. A závěrečné potvrzení a ukončení celé transakce, STRÁNKA 7.

Tento celkem jednoduchý on-line nákup se skládal celkem ze sedmi zpátečních cest mezi uživatelem a serverem, což je právě typické chování webových aplikací. Na desktopu je podobná interakce otázkou několika vteřin. Avšak na webu? Pokud se vezme v potaz, že připojení k síti a odezvy ze stran serveru jsou pomalé, může každý z těchto kroků trvat třeba i několik sekund. Výsledek? Zákazník ztrácí trpělivost a odchází jinam. Třeba ke konkurenci.

Na základě předchozího příkladu, je třeba uznat, že pro složitější aplikace, které vyžadovaly svižnou komunikaci mezi klientem a serverem, byl tento stav opravdu tristní. Pokud vezmeme v potaz, že mnoho webových aplikací mělo ambice stát se alternativou k desktopovému softwaru, tak nemohly na tomto modelu stavět. Bezespору musela přijít nová řešení, která by tyto nedostatky překlenula.

Zpracováno na základě informací z [1]

4 Co je Rich Internet Application

V návaznosti na předchozí kapitolu, Rich Internet Applications představují právě ta nová řešení jak překlenout problémy vycházející ze samotné povahy webu.

Konkrétně u uváděného e-shopu, kdy zákazník při každé nové události musí čekat na odezvu serveru, RIA toto nežádoucí chování eliminuje. Veškerá interakce mezi aplikací a uživatelem probíhá na klientské straně. A se serverem komunikuje pouze v případě výměny důležitých dat, jenž tak není zatěžován banálními úkony pro posílání každé stránky. Komponenty uživatelského rozhraní se na stranu klienta načtou jen jednou (nebo se inteligentně mění po částech), tedy odpadá proces opakovaného překreslování v prohlížeči. To má za následek, že GUI může být daleko bohatší na grafiku, efekty či media, tak jak jsou lidé zvyklí z desktopu. Výsledkem je větší spokojenost uživatele, protože práce s programem se stává pohodlnější a svižnější.

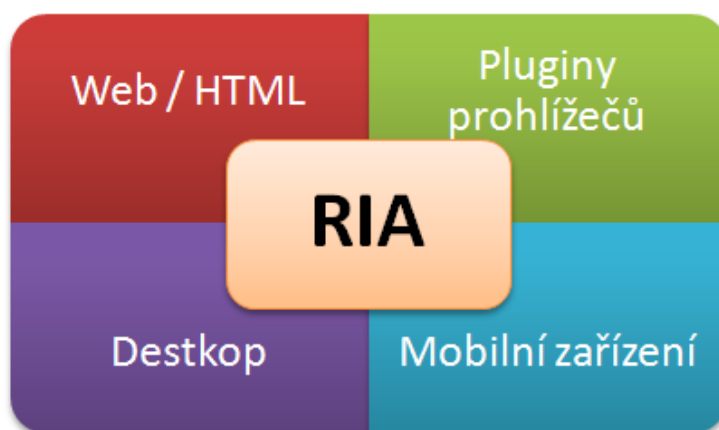
Jak by se tedy RIA daly výstižně charakterizovat?

„RIA (Rich Internet Application) jsou webové aplikace, které se snaží překlenout rozdíl mezi klasickou webovou a desktopovou aplikací. RIA aplikace se snaží v rámci webového prohlížeče napodobovat desktopové aplikace svým vzhledem i chováním a poskytnout vyšší uživatelský komfort.“ [32]

„Aplikace, které kombinují širokou dostupnost webu s uživatelským komfortem desktopových aplikací.“ [11]

Rich Internet Application kombinuje výhody používání webu jako nízkonákladového modelu pro umístění aplikací s bohatým uživatelským zážitkem přinejmenším tak dobře jako dnešní desktopové aplikace. [1]

Jak je vidět, charakteristik je možné nalézt mnoho, a vesměs se opírají o stejnou myšlenku. Nicméně problém pro přesnou konkretizaci, co vlastně je RIA, představuje pole jejich současného nasazení. „Díky expanzi na desktop a do mobilních zařízení se však sám pojem "webová aplikace" dostává do určité krize, proto je dnes vhodné RIA považovat spíše za směr než za něco, co by šlo exaktně definovat.“ [11]



Obrázek 4. Současné pole působnosti RIA

4.1 Teoretický základ

RIA jakožto jméno pro nové řešení se objevilo v roce 2002 ve společnosti Macromedia, respektive vděčí za svůj vznik jejímu tehdejšímu technologickému řediteli Jeremy Allairovi. Ten v souvislosti s vydáním další verze programu Flash o tom pojednává ve svém white-paperu, ve kterém předpovídá budoucnost ve vývoji webových aplikací. Přesto, že se jedná už o 7 let starý koncept, v současnosti je více než aktuální.

Od roku 2002 bude Internet jiný. Koncový uživatelé a firmy vyžadují více od svých investic do internetových technologií. Schopnost doručit uživateli skutečnou hodnotu nutí mnoho společností se přiklánět k bohatším modelům internetových aplikací. Modely, které kombinují vlastnosti tradičního desktopu s umístěním a bohatým obsahem webových aplikací.

Společnosti také očekávají vzrůst v používání webu jako služby nebo opětovné použití softwarových komponent, které budou používány napříč celou sítí. Dále budou vyžadovat prostředí, kde aplikace budou muset sdílet funkcionalitu a data napříč mnoha druhy koncových zařízení.

Tyto trendy dávají našemu snažení směr k nové generaci bohatých klientů. To je perspektiva, pro kterou Macromedia vyvinula Flash MX a Flash player 6. [13]

Po úvodní kapitole, kde Jeremy Allaire nastínil poměrně pravdivou vizi, jak to vlastně s Internetem bude vypadat, pokračuje v obecné charakteristice vlastností budoucích webových technologií. Jeho názory jsou jednoznačně ve prospěch softwaru Macromedie a nezakrývají silnou kritiku vůči prohlížečům a HTML, proto je nutné je brát s určitou rezervou. Avšak autor bakalářské práce pokládá za vhodné tyto myšlenky uvést, jakožto ucelený teoretický základ, co by měli technologie či jednoduše RIA splňovat a umět.

- I. Měly by poskytovat efektivní výkonné běhové prostředí pro zpracování kódu, obsahu a komunikace mezi aplikací a uživatelem. Tím je myšleno zajištění funkce odstínění modelu žádost/odpověď a tak docílit redukce opakovaného přenosu základních dat související s uživatelským rozhraním. Pro tyto potřeby je třeba na klientské straně zajistit datové uložště, ze kterého by byla poskytována aplikační logika a základní informace pro grafické zobrazení.
- II. Integrovat obsah, interakci s uživatelem a aplikační rozhraní do společného prostředí. To znamená, že současný uživatel vnímá prezentační vrstvu rozděleně. Tento požadavek lze brát jako kritiku na HTML, který primárně zajišťuje jen textový obsah a postrádá prostředky pro přehrávání zvuku, videa a dalších medií. Ty musí mít na starosti jiné konkrétní aplikace. Bohatý klient by měl prezentaci všech těchto forem obsahu poskytovat sám.
- III. Poskytnout silnější a více flexibilní objektový model zajišťující interaktivitu, než který poskytují současné webové prohlížeče. Jejich Současná podpora DOM, Javascriptu a DHTML není dostatečné řešení pro budování opravdových Rich Internet Applications. Je třeba dobře fungujícího objektově založeného modelu pro budování aplikací z hlediska UI nebo komunikace s uživatelem.

- IV. Umožňovat rychlý vývoj aplikací za pomoci znovupoužitelných komponent. Vývoj by měl usnadnit používání vizuálních a funkčních částí, které jdou snadno integrovat do daného projektu bez ohledu na to, jestli jejich autorem je daná firma, která aplikaci vytváří nebo je poskytnete třetí nezávislá strana. Vnikl by se tak základ pro rychlý a efektivní vývoj.
- V. Užívání webových a datových služeb poskytovaných serverem. V RIA jsou jednoznačně odděleny vrstvy uživatelského rozhraní, prezentační logiky od logiky aplikační. Klientská část aplikace by měla být schopna poskytovat vzdálené služby aplikačního serveru nebo dalších webových služeb.
- VI. Schopnost funkce v režimu on-line i off-line. Aby mohl uživatel s aplikací pracovat neměla by být její funkcionalita jednoznačně závislá na stálém připojení k Internetu. Těchto výhod využijí zejména mobilní zařízení, u kterých kvalita a dostupnost připojení je často variabilní na základě síly signálu a pokrytí.
- VII. Dostupnost na co největším spektru platform a zařízeních. RIA by měla fungovat na všech osobních počítačích a nejrozšířenějších operačních systémech stejně tak dobře jako na dalších zařízeních jako Smartphones, PDA, set-top-boxech nebo herních konzolích.

Kapitola zpracována dle [13]



Obrázek 5. Modelový elektronického obchodu, který pracuje v prostředí Flash Playeru.

5 Přehled dostupných technologií

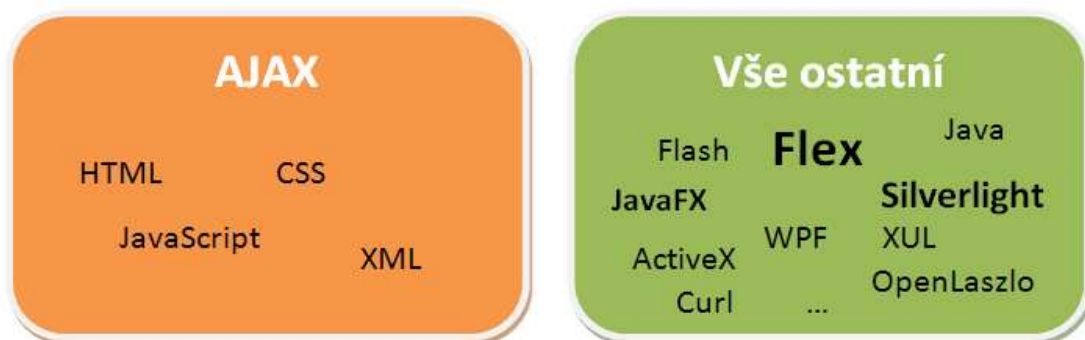
Jedny z prvních pokusů, které předznamenaly RIA přišly s nástupem appletů, které běželi v prohlížeči prostřednictvím Java Virtual Machine. Nikdy se však v tomto ohledu nestaly populární a Java našla své uplatnění především na serverové straně a později u mobilních zařízení.

Po přelomu milenia se na poli RIA technologií začalo prosazovat hned několik hráčů. Podle jejich technického charakteru by bylo vhodné rozdělit do dvou velkých skupin.

První skupinu tvoří pouze jediný zástupce Ajax, který představuje pokračovatele klasických zavedených webových technologií jako JavaScript, HTML a CSS, které jsou interpretovány samotným prohlížečem. V jednom článku se autor dokonce setkal a charakteristikou „Ajax versus zbytek světa“, která poměrně vtipně vystihuje jeho specifické postavení.

Druhá početnější skupina je právě ta, která představuje ten „zbytek světa“. Její zástupci se často od sebe výrazně liší, nicméně je spojuje princip existence jejich běhového prostředí (runtime), což bývá nejčastěji nějaký plugin nainstalovaný ve webovém prohlížeči. Pro formálnost je možné je označovat jako technologie „ostatní“, nebo pluginové.

Dle informací zveřejněných na [11]



5.1 Ajax *Množiny jednotlivých technologií*

Hned na úvod je potřeba říci, že Ajax není považován za technologii jako takovou, ale spíše za techniku, či návrhový vzor pro využití dosavadních webových technologií.

O některých již bylo napsáno v úvodí kapitole Historický vývoj a některé je třeba ještě představit. Podrobnější charakterizací lze Ajax označit jako komunikační model, který využívá kombinaci těchto prvků: Document Object Model (DOM), XMLHttpRequest (XHR), (X)HTML, CSS a JavaScript

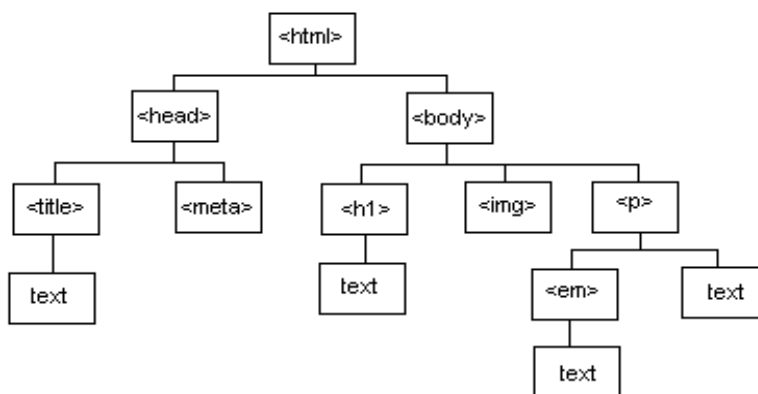
Právě pro JavaScript se nasazení Ajaxu stalo spásou, kterou probudil zájem o jeho opětovné používání. Jako termín se poprvé objevil v roce 2005 ve článku Jesse Jamesa Garretta s názvem Ajax: A New Approach to Web Application. Garret pojednává o tom jak se zmírňuje rozdíl mezi aplikacemi na webu a desktopu, právě za použití výše zmiňovaných prvků. Termín Ajax vymyslel jako zkratku několika vhodně zvolených složek, které zajišťují jeho typické fungování. Tedy **A**synchronous **J**avascript **A**nd **X**ML, což je mnohem elegantnější a zvučnější, než tehdy obecně uváděná kombinace XMLHttpRequest, DOM, JavaScript, XML nebo HTML a CSS. I přes tuto jakousi popularizaci, opravdový zvuk Ajaxu daly až společnosti jako Google, Yahoo! nebo Amazon, které ho implementují ve svých aplikacích. Pro jasnější pochopení bude následovat několik kapitol, které popisují funkcionalitu jednotlivých komponent.

Zpracováno podle [10]

5.1.1 Document Object Model

Document Object Model (DOM) je způsob, kterým webové prohlížeče interpretují HTML dokument jako strukturu objektů JavaScriptu. Jednotlivé prvky HTML dokumentu jako odstavce, záhlaví nebo texty jsou prezentovány jako jednotlivé objekty, ke kterým může programátor jednotlivě přistupovat prostřednictvím funkcí JavaScriptu a měnit jednotlivé části dle potřeb. Navíc je poskládá do stromové struktury, takže mezi objekty funguje vztah rodič-potomek, což usnadňuje jednodušší navigaci v dokumentu.

Na základě [4]



Obrázek 7. Strom jednotlivých elementů DOM

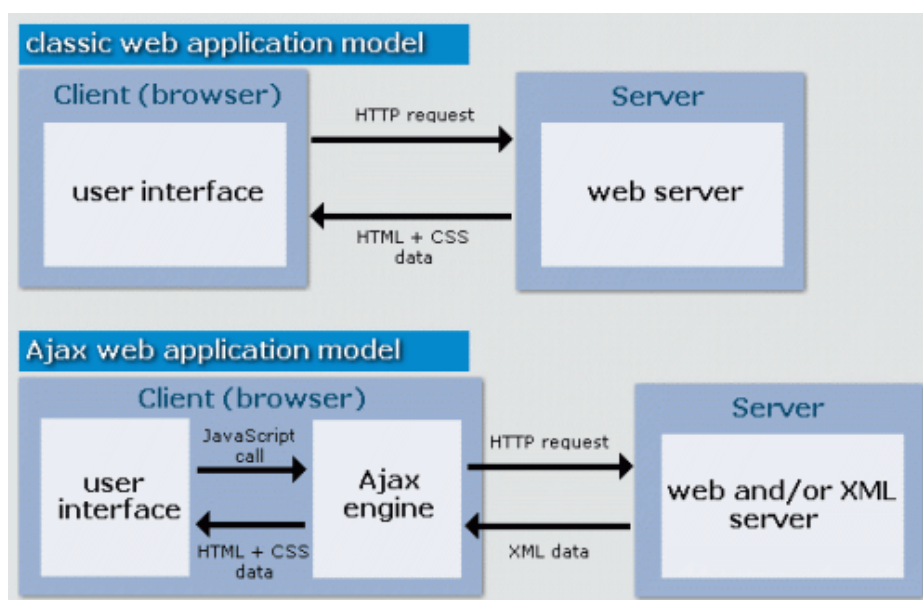
5.1.2 XMLHttpRequest

Možná nesrozumitelně vyhlížející slovo XMLHttpRequest (XHR) právě reprezentuje označení pro klíčový prvek Ajaxu, který tak podnítl jeho velký rozmach.

S pohledem do nedávné minulosti nepředstavuje nic nového. Jeho první podporu zavedl už v roce 1999 Internet Explorer 5 jako ovládací prvek Active X, což je softwarová komponenta, která umožňuje určitou opakovanou funkcionalitu. Takové komponenty umožňují skriptům na straně klienta přístup k výbavě operačního systému a v případě XHR se jedná o přístup k síťové vrstvě. Uznávaným webovým standardem se stal až po rozšíření na ostatní prohlížeče s příchodem Mozilly 1.0 a Safari 1.2.

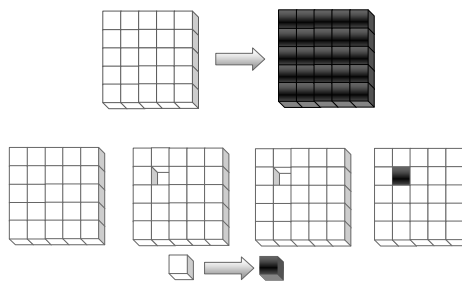
XHR tedy představuje funkcionalitu webových prohlížečů, která umožňuje JavaScriptu zavolat server, způsobem, jenž obchází klasický model žádost - nová stránka. Jedná se o asynchronní volání serveru (**Asynchronous** Javascript and XML) respektive asynchronní požadavky. V praxi to znamená, že když uživatel prohlíží aktuální stránku, JavaScript na pozadí provádí dodatečné požadavky na server. Dohromady s DOM tak může aktualizovat nebo vymazávat jen část UI aniž by musela být překreslena celá stránka.

Ná obrázku číslo. 8 je znázorněn klasický a Ajaxový model. Je patrné, že u Ajaxu přibyla další vrstva Ajax engine, což je jakási prezentační logika napsaná v JavaScriptu, která obsluhuje objekt XHR. Odděluje tak přímou komunikaci klientského UI od webového serveru, který jí potřebná data nejčastěji posílá přes XML.

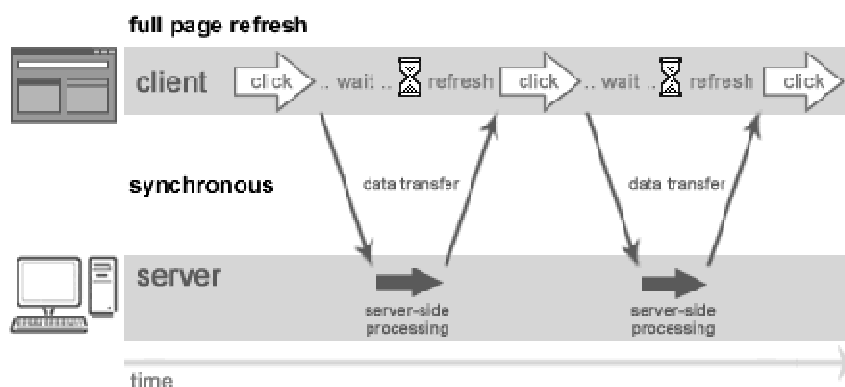


Obrázek 8. Klasický a ajaxový webový aplikační model

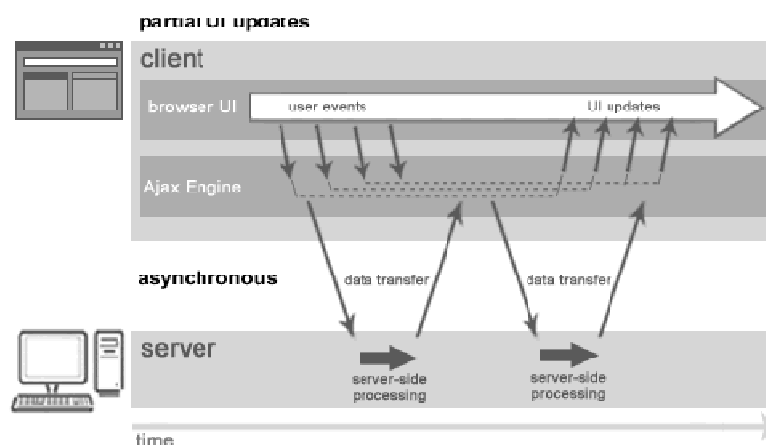
Zpracováno podle [10]



Obrázek 9. Normální a ajaxový požadavek stránky



Obrázek 10. Synchronní komunikační model



Obrázek 11. Asynchronní komunikační model.

5.1.3 Knihovny a aplikační rámce

Dříve nebo později se při vývoji aplikací každý programátor setká s tím, že vytváří kód pro realizaci stále stejných konstrukcí, které se rutinně opakují. Pokud dodržuje pravidla tří vrstev lehce znovu použije kód vlastní nebo ještě lépe si může sestavit celou vlastní knihovnu. Právě volně dostupných JavaScriptových knihoven, které by měli vývojářům usnadňovat práci, je dnes už celá řada. Některé jsou svým rozsahem malé

a specializují na konkrétní problémy, na druhé straně vnikají rozsáhlé knihovny, které se neustále rozšiřují o další funkcionality ve snaze postihnout, co největší šíři interakce, která je jen na webu možná. Na tomto základě vnikají tak celé aplikační či pracovní rámce (frameworky), které by tak měly tvořit zázemí pro vývoj webových aplikací.

Mimo jiné se v designu uživatelského rozhraní najde mnoho zažitých standardů, na které je uživatel zvyklý. Ví, jak pracovat s posuvníky, rozbalovacím menu nebo navigací po mapě. Vymýšlet něco originálnějšího a porušovat tak tyto konvence by bylo většinou na škodu a uživatele by jediné mátl. Některé interakce jsou tedy natolik běžné, že se vyplácí využívat znovupoužitelných komponent tzv. widgetů, ze kterých se výsledné UI sestaví. Zajišťují veškerou užitečnou funkcionalitu a jsou opět obsaženy v celých knihovnách. Hovoří se tak o pracovních rámcích komponent (component frameworks), které zapouzdřují již samostatnou implementaci Ajaxu.

Společnou výhodou práce s těmito frameworky je vysoká úroveň abstrakce při samotném psaní programu a také již odladěná kompatibilita pro jednotlivé druhy webových prohlížečů. Vývojář tak nemusí aplikaci stavět na „zelené louce“ a namísto psaní stovek řádků HTML, CSS a JavaScriptu, vloží danou komponentu, což se často rovná jedné řádce napsaného kódu. To mu bezpochyby umožní soustředit se lépe na podstatu aplikační logiky, protože se nemusí zabývat implementačními detaily.

Pokud pomineme přímo Ajaxové frameworky, které jsou vhodné spíše pro webové stránky, tak na vývoj RIA jsou převážně nasazované právě frameworky komponentové. V současné době se vyskytují v poměrně vysokém počtu a většina je vyvíjena jako open-source, jen pár jich je proprietárních. Vzhledem k jejich různorodé kvalitě a rozšíření budou na následující straně uvedeny jen ty nejznámější.

- **Dojo**

Patří k jedním z nejstarších aplikačních rámců, byl představen poprvé v roce 2004. Pro rychlý vývoj je k dispozici více než stovka komponent (roztahovací nabídky, pole se seznamem, stromové nabídky či ukazatele průběhu), které se jednoznačně zaměřují na funkcionalitu.

- **Rico**

Název Rico znamená ve španělštině bohatý, což je jasné vysvětlení na co je především určen. Narozdíl od Doja hlavní silou Rica je zaměření na techniky drag&drop, datových polí a efektní animace.

- **Yahoo! UI Libary**

Známý internetový vyhledávač Yahoo! uvolnil své knihovny a představuje tak framework pro snadnou práci s Ajaxem, respektive DHTML, JavaScriptem, CSS a DOM. Podléhá licenci BSD.

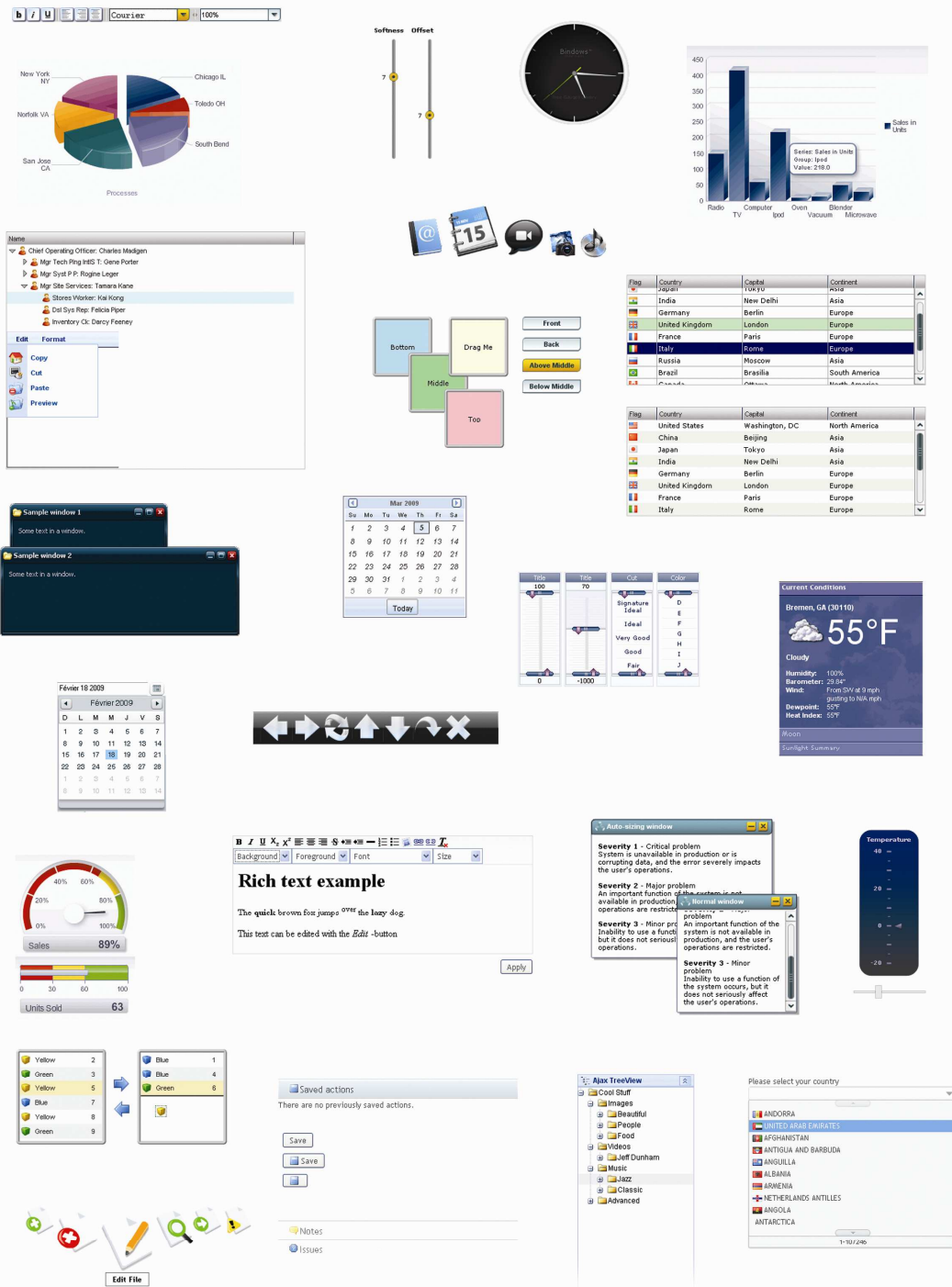
- **Oracle ADF Faces a ADF Data Visualization**

I společnost, která se zaměřuje na databázové systémy si uvědomila jako sílu představuje bohatá klientská aplikace. Vyvinula tak pro své potřeby řadu Ajaxových komponent, které jsou k dispozici v rámci jejich vývojového nástroje JDeveloper 11g. ADF Faces představují hotové navigační prvky GUI a ADF Data Visualization zahrnují oblast interaktivních grafů a map.

- **Google Web Toolkit (GWT)**

Samozřejmě taková společnost jako Google, která na Ajaxu postavila mnoho svých internetových aplikací, bude disponovat vlastním aplikačním rámcem. GWT používá pro svůj základ Javu, takže je programátor odpoután od znalostí Ajaxových technologií. Stručně řečeno aplikace se píše v Javě a ta se výsledně zkompiluje do HTML a JavaScriptu, které je optimalizované pro většinu známých prohlížečů.

Zpracováno na základě [4] a [5]



Obrázek 12. Ukázka několika komponent jednotlivých aplikačních rámců

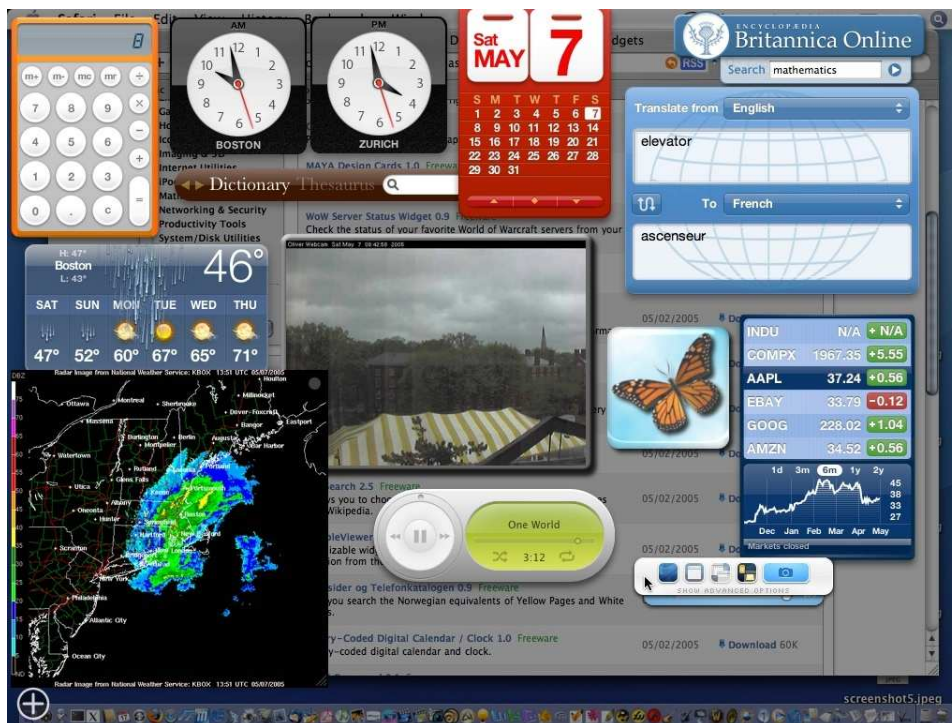
5.1.4 JavaScript mimo web

Vzhledem k tomu, že znalostí JavaScriptu začalo disponovat poměrně velká komunita softwarových vývojářů, začal být tento jazyk používán i pro vývoj aplikací a komponent mimo web. Názorným příkladem toho jsou Dashboard Widgets, které se poprvé objevili u operačního systému Mac OS X.

Stejně jako v případě frameworků je widgetem míněna nějaká funkční komponenta. Avšak v přenesené souvislosti je její význam brán jako miniaplikace na ploše operačního systému, které se specializuje na jedinou funkci. Nejčastěji to bývá automatické získávání aktuálních informací z Internetu, aniž by uživatel musel otevírat webový prohlížeč. Existují tak widgety zobrazující předpověď počasí, burzovní zprávy, světový čas, kalendář nebo cizojazyčné slovníky.

Ve své podstatě nepředstavují nic jiného než zlomky HTML, CSS a JavaScriptu a jejich snadné programování zapříčinilo obrovský nárůst těchto komponent, kterých je dnes na síti dostupných více jak tisíc. Podobné prvky byly zařazeny i do Windows Vista, pod odlišným označením gadgets, které de-facto vyjadřuje to samé.

Závěrem za zmínku stojí, že prostřednictvím JavaScriptu a XUL, (XML User Interface Language) je vyvinutý celý internetový prohlížeč Firefox.



Obrázek 13. Dashboard Widgets na Mac OS X

Dle informací z [4] a [5]

5.2 Adobe Flash Platform

V současné době asi nejznámější pluginovou technologií představuje Flash a od doby, kdy se poprvé objevil, ušel již poměrně dlouhou cestu vývoje. Macromedia rozšiřovala svůj produkt o další funkce a možnosti, takže z jednoduchého animačního programu se vyvinul poměrně silný nástroj, který je schopen vytvářet i webové aplikace.

Nelze zpochybnit, že Flash změnil tehdejší paradigma na web jako takový. Internet prvotně chápaný jako prostředí pro prezentaci textu se s jeho příchodem otevřel interaktivní bohaté grafice, na který klasický model na bázi HTML nestačil. Stal se z něj univerzální nástroj se širokým polem uplatnění, který si vybudoval silnou komunitu tvůrců napříč profesním spektrem, o čemž svědčí mnoho fandovských webů zaměřených na oblasti animace, webdesignu nebo programování.

S příchodem další verze Flash MX se společnost orientuje na nový koncept webových aplikací s bohatším obsahem a představuje nové nástroje, které usnadňují jejich tvorbu. Zásadním okamžikem se stává rok 2005. Macromedii pohltila konkurenční společnost Adobe Systems a dala si za úkol plně integrovat její produkty mezi ty své. Vnikla tak řada programů a technologií zaměřených na vývoj Rich Internet Applications, označovaná jako Adobe Flash platform. [7]



Obrázek 14. Nástroje a technologie platformy Adobe Flash.

5.2.1 Flash Player

Flash Player je zásuvný modul do internetového prohlížeče, který je určený k přehrávání souborů SWF. Těmito soubory mohou být animace, videa, on-line hry nebo klienti webových aplikací.

Dnes se nachází ve verzi 10 a má svou alternativu i pro mobilní zařízení s označením Flash Lite. Jeho distribuce je zdarma a v současnosti je podporován nejrozšířenějšími prohlížeči a operačními systémy. Adobe uvádí, že je nainstalován na 99% počítačů připojených k Internetu, což ho řadí mezi nejrozšířenější webový přehrávač vůbec.

Vzhledem k tomu, že musí být integrován do nějakého prohlížeče, nastává tak paradoxní situace, kdy uživatel spouští aplikaci - prohlížeč, ve kterém běží aplikace druhá – plugin, ve kterém by následně měla fungovat nějaká aplikace třetí.

Webový prohlížeč tak degradoval na pouhého nosiče, přičemž veškerá schopnost prezentace dat a funkcionalita je jen v možnostech Flash Playeru. Vzhledem k tomu, že se jedná o přehrávač velice univerzální, disponuje poměrně širokou nabídkou funkcí. S ohledem na téma a rozsah bakalářské práce bude tedy věnována pozornost jen těm zásadním, které souvisí s podporou aplikací a dále zajímavým novinkám, které přišly s verzí 10.

Prostředí pro chod aplikací

Asi nejdůležitějším aspektem je programovací jazyk. Flash využívá jazyk ActionScript, jehož první náznaky se objevily již v ranných verzích programu a poskytovaly omezené příkazy pro ovládání časové osy jako "play", "stop" nebo "goto". V současnosti má za sebou přes 10 let vývoje a představuje slušný objektově orientovaný programovací jazyk. Jeho aktuální verze 3.0. vychází ze standardizovaného jazyka JavaScript, již zmiňovaný ECMAScript. Kód ActionScriptu je kompilován do bytecode (podobně jako Java), který je uložený do souboru SWF. Samotná interpretace probíhá v ActionScript Virtual Machinu, který je součástí Flash Playeru. Překládá bytecode do nativního kódu počítače, což zajišťuje maximální rychlost jeho provádění.

Co se týče manipulací a výměnou dat, tak v návaznosti na předchozí kapitoly věnované Ajaxu je třeba zmínit, že Flash Player taktéž umožňuje asynchronní komunikačním modelu, dříve označován jako Afax (Asynchronous Flash and XML).

Podstatnou novinkou je, že Flash aplikace mohou přímo nahrávat a ukládat soubory na klientském počítači nebo je vzájemně sdílet. Podporován je i bezpečný přístup ke schránce pro funkce vkládání textu nebo lokální uložiště společných dat do mezipaměti přehrávače. Komplexně se tedy jeho prostředí přibližuje tak, aby umožnilo uživateli řádnou práci s aplikací, tak je zvyklý na desktopu.

Prezentace grafiky a multimedií

Obecně vzato jsou pluginy pro oblast prezentace grafického obsahu mnohem lépe vybavené než klasické HTML. Flash Player v sobě zahrnuje použití vektorové a bitmapové grafiky, animaci, funkce přehrávání videa, audia, streaming rozšířený o dynamický streaming. S verzí 10 přišla podpora 3D, na kterou se u Flashe dlouho čekalo, tvorba vlastních filtrů, různých efektů a rozšířená práce s textem, u které Adobe uplatnilo své bohaté zkušenosti z DTP. A nakonec zcela zásadní novinkou je využití výpočetního výkonu hardwaru grafické karty pro vykreslování souborů SWF.

Více na [22]

5.2.2 Flash

Ačkoliv mnoho „zatvrzelých programátorů“ nebude souhlasit, autor bakalářské práce je toho názoru, že na přidané hodnotě software se bezesporu projevuje i kvalita grafického uživatelského rozhraní. A zejména pro webdesign a současně pro RIA to platí dvojnásob. Právě proto následující text patří nástroji Flash, který je primárně chápán jako grafický software. To kdy a jakým způsobem se poprvé objevil bylo probráno v kapitole věnované historii, ale zásadní otázkou je, co může nabídnout webovým aplikacím.

Ve zjednodušeném pohledu Flash představuje vhodnou kombinaci několika elementů. To jest práce s grafikou, animace a programování. Jeho základní uživatelské prostředí tvoří vektorový editor, který obsahuje kreslicí nástroje s podporou pro práci s textem. Vytvořený grafický koncept lze animovat prostřednictvím znázorněné časové osy. Veškeré tato práce je na bázi klikání myši na příslušná tlačítka funkcí. Návrhář tak může vytvářet dynamickou webovou grafiku, například tak jako pomocí JavaScriptu, aniž by znal jakýkoliv skriptovací jazyk. Jedná se o takzvaný WYSIWYG mód (akronym anglické věty *What you see is what you get*“, „Co vidíš to dostaneš“), který je hlavně podporován řadou HTML editorů jako Adobe Dreamweaver nebo dřívější Microsoft FrontPage. V tomto případě jde o funkce jednoznačně zaměřené na grafické desingéry a animátory, u kterých se nepředpokládá hluboká znalost nějakých programovacích jazyků.

Nicméně, pokud tvůrce nechce zůstat pouze u animací, programování se nevyhne. K těmto potřebám má Flash implementovaný, již zmiňovaný, jazyk ActionScript. Veškeré vizuální prvky a jejich dynamika jdou ovládat pomocí objektového modelu tohoto jazyka. Mnoho jeho tříd a definovaných funkcí umožňují naprogramovat všechnu interaktivitu, přehrávání, prezentaci a vykreslování dat. Finální grafický koncept vybavený takto naprogramovanou logikou je pak kompilován do souboru SWF.

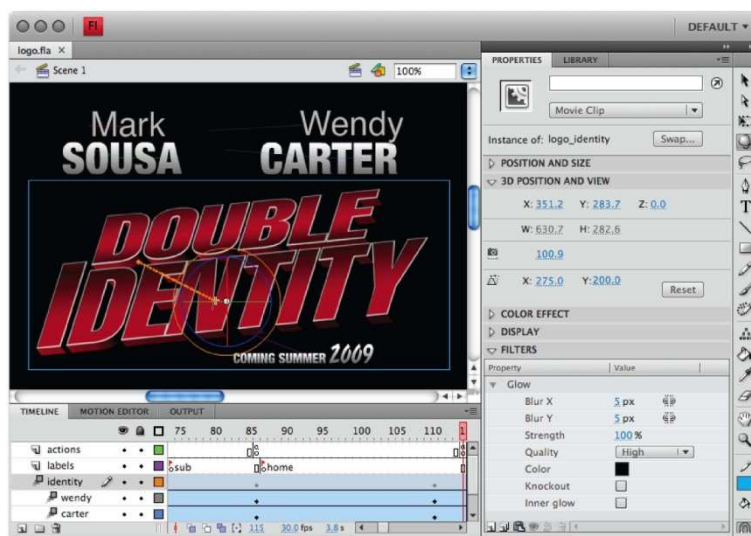
Právě kombinace atraktivní grafiky a dostupnost programovacích prostředků z Flashe udělali populární nástroj pro tvorbu on-line her a multimediálních webových prezentací, od kterých logicky vedla cesta až k oblasti Rich Internet Applications.

Jak je tedy Flash pro RIA vybaven? Ačkoliv se jedná o autorské prostředí má k dispozici knihovnu už hotových komponent uživatelského rozhraní, které lze upravovat pomocí grafického editoru nebo přímým kódováním v jazyce ActionScript. Pro psaní skriptů je k dispozici řada pomocných nástrojů jako skriptovací asistent pro automatické dokončení syntaxe, ladící program nebo profil vývoje pro mobilní zařízení. Action Script umí pracovat s jazykem XML a dorozumí se v rámci prohlížeče i s JavaScriptem.

Momentálně nejaktuálnější verzí je Flash CS4 Professional, který opět přinesl řadu vylepšení. V poslední době je kladen důraz na silnou spolupráci s ostatními produkty Adobe a řada jejich funkcí je na Flash přenášena, tak aby ve svém rámci poskytoval co nejvíce tvůrčích možností. Nicméně stále se jedná především o grafický a animační software a jeho poslední rozšíření přineslo nové funkce ohledně 3D grafiky, modelování a kinematiky.

Autor je toho názoru, že RIA je pro Flash současným zenitem jeho maximálního využití. Vývojáři bohatých aplikací se tak dostali do prostředí na pomezí grafiky a programování, které pro ně není často ideální. Své místo si bude nadále držet v oblasti reklamy související s vytvářením bannerů, atraktivních prezentací a mnohokrát zmiňované animace. Pro účely skutečného vývoje slouží už jiné prostředky, kterým bude věnována následující kapitola.

Více na [22] a [28]



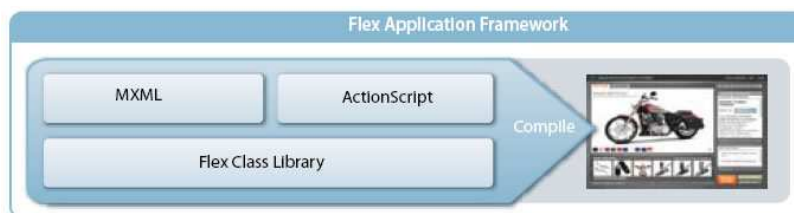
Obrázek 15. Aplikace Adobe Flash

5.2.3 Flex

Jakmile bylo možné pomocí nástroje Flash vytvořit bohatou webovou aplikaci začala být tato platforma velice atraktivní. A jak už bylo řečeno, Flash se svojí koncepcí určenou primárně pro grafiky nedostačoval potřebám vývojářů a vzbuzoval nedůvěru k jeho použití. Na tomto základě Macromedia vybudovala Flex, soubor vývojových technologií postavených nad Flash Playerem. Oproti svému předchůdci neobsahuje časovou osu ani prvky pro vytváření grafiky. Flex tedy není určený pro autorskou grafickou tvorbu, ale nabízí snadný vývoj webových aplikací, při kterém programátoři využívají své zažité pracovní postupy.

Jeho obecný koncept velice připomíná model HTML/JavaScript. Základ tvoří jazyk MXML(Maximum eXperience Markup Language), speciální značkovací jazyk na základech XML, který slouží pro definici a výstavbu uživatelského rozhraní. A samozřejmě ActionScript 3.0 pro programování části aplikační logiky. Dochází zde k jednoznačnému separování prezentační vrstvy od samotné funkcionality aplikace, což usnadňuje jednotlivé fáze vývoje. Grafický designér tak může odděleně navrhovat vzhled GUI pomocí MXML, zatímco programátor pracuje se skriptovacím jazykem. Stejně tak probíhá proces zpětné editace aniž by si vzájemně zasahovali do svého kódu.

Ve finále jsou oba dva jazyky kompilovány opět do souboru SWF, který zcela odpovídá formátu generovaným Flashem. To se zcela odlišuje od klasického vývoje webových aplikací, kde ke kompilaci nedochází a interpretem je internetový prohlížeč. Zásadní výhodou je, že pro napsání a úpravu zdrojových kódů není potřeba žádný speciální software jako v případě Flashe, který má své soubory uložené v binárním formátu.



Obrázek 16. MXML a ActionScript se zkompilují do souboru SWF

Čili k vývoji v MXML a AS 3.0 postačí jakýkoliv textový editor s dostupným SWF kompilátorem. Adobe k těmto účelům uvolnila open source framework Flex SDK. Nicméně pro rychlou a efektivní práci je vhodné využít komerční verzi Adobe Flex Builder. Jedná se o prostředí založené na softwaru Eclipse., který má všechny vymoženosti k usnadnění vývoje jako je inteligentní programování, krokování při ladění chyb nebo vizuální návrh GUI(WYSIWYG).

Tak jako v případě Ajaxových frameworků, tak i Flex Builder obsahuje aplikační rámec s více než 100 předpřipravenými komponentami pro budování prvků uživatelského rozhraní jako tlačítka, zaškrtačací pole, posuvníky, rolovací nabídky atd. Jejich vzhled se upravuje styly na bázi CSS nebo připravenými skiny. Do struktury grafických prvků však nelze přímo zasahovat, proto mají Flexové aplikace nezaměnitelný vzhled.

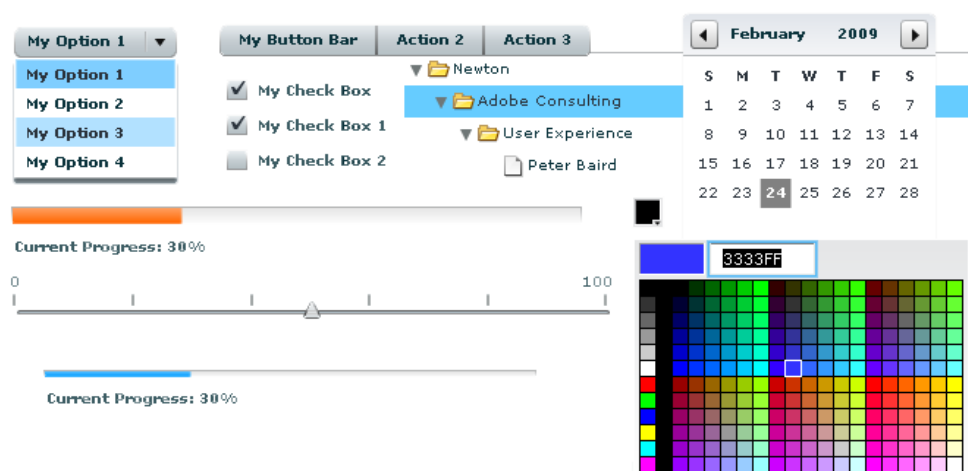
Co se týče nasazení na server, tak je zde postup téměř totožný jako při publikování dokumentů v HTML. Zkompilovaný soubor SWF stačí nahrát na server, který pak po načtení na klientskou stranu s ním komunikuje skrze HTTP.

Zpracováno na základě [1], [14], [15]

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <mx:Application xmlns:mx="http://www.adobe.com/2006/mxml" layout="absolute">
3
4 <mx:Script>
5   <![CDATA[
6     import mx.controls.Alert;
7
8     private function sayHello() : void {
9       Alert.show("Ahoj lidi!");
10    }
11  ]]>
12 </mx:Script>
13
14 <mx:HBox x="83" y="54">
15   <mx:TextInput id="textBox" />
16   <mx:Button label="Klikni!" click="sayHello()"/>
17 </mx:HBox>
18 </mx:Application>
  
```

Obrázek 17. Ukázka zdrojového kódu Flex aplikace



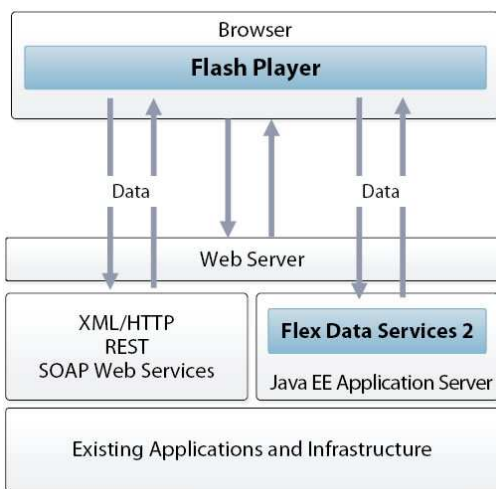
Obrázek 18. Flex komponenty

5.2.4 OpenLaszlo

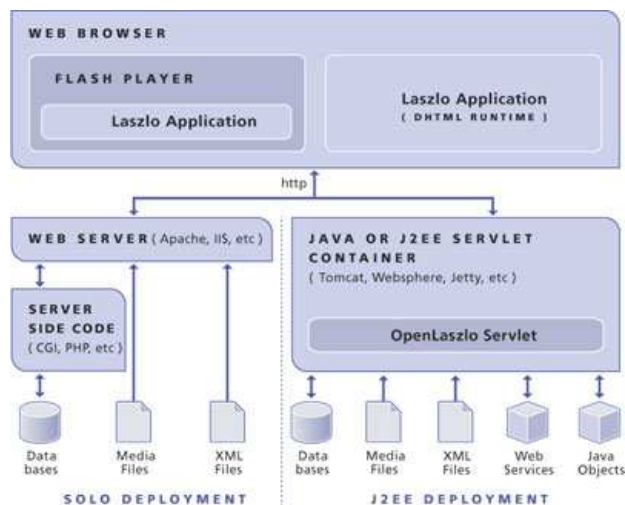
Dalším frameworkem pro tvorbu bohatých internetových aplikací, které fungují v prostředí Flash Playeru představuje OpenLaszlo. Jedná se o open source framework s komerční podporou firmy Laszlo Systems, který nabízí velmi podobný vývojový model jako má Flex.

Jednotlivé komponenty GUI jsou definovány v jazyce LZX, který opět vychází ze základů XML, a o aplikační logiku se stará JavaScript. Výsledná aplikace pak může být generována do DHTML nebo zkompileována do souboru SWF. Schopnost generovat DHTML kód dělá z Laszla slušného kandidáta na vývojářské prostředí pro mobilní aplikace o čemž svědčí spolupráce Laszlo Systems a Sun Microsystems, který se snaží umístit tuto technologii pro svou Java Mobile.

Za zmínku stojí, že jak Flex tak Laszlo - na rozdíl od Ajaxu - nabízejí řešení i pro stranu serveru. Mimo využití serverů společností třetích stran mohou využívat i proprietární serverové části, které mohou udržovat stavy jednotlivých klientských komponent nebo zajišťovat komunikaci v datově efektivním formátu.



Obrázek 19. Architektura Flex aplikací



Obrázek 20. Architektura Laszlo aplikací

Více informací k nalezení na [1], [22], [23]

5.2.5 AIR

V roce 2008 Adobe představila novou kategorii běhového prostředí umožňující chod webových aplikací v režimu off-line. Jedná se o modul AIR (Adobe Integrated Runtime), původně vyvíjen pod označením Apollo. Řeší zásadní problémy související s přístupem k systémovým prostředkům na klientském počítači, které má Flash Player z bezpečnostních důvodů omezené. AIR je však schopen využívat všech výhod desktopu jako je práce se soubory, přístup k databázím nebo k operačnímu systému.

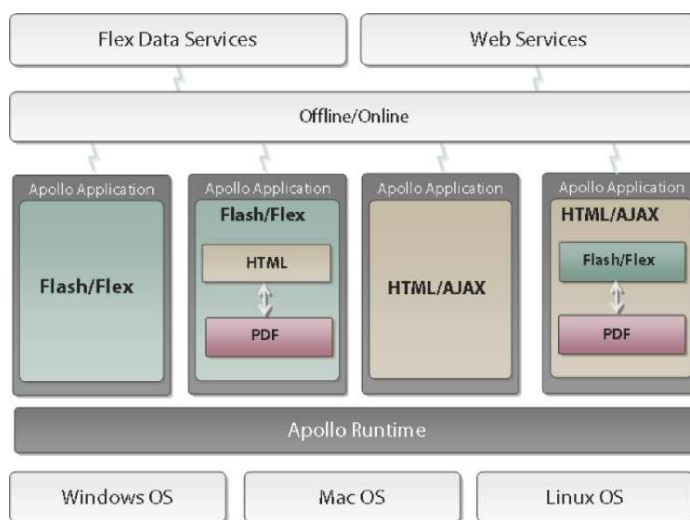
Internetové aplikace se tak paradoxně přesouvají mimo webové prohlížeče a vyplňují pomyslnou mezeru mezi desktopovým programem a webovou službou.

Nabízí se otázka, zda-li se jedná o tak zásadní počin, vzhledem k tomu, že pomocí JavaScriptu jdou vytvářet desktopové widgety (kapitola 2.1.3). Dalo by se říci, že v tomto případě se Adobe drží podobné koncepce avšak s velkým důrazem na multiplatformnost. AIR je aktuálně spustitelný na systémech Windows, Mac a Linux, takže garantuje nezávislost na jednotlivých platformách, což řeší mnoho problémů souvisejících s kompatibilitou.

Neméně důležitým aspektem je, že AIR aplikace mohou být vytvářeny pomocí HTML, CSS, JavaScript, Flash, Flex nebo za použití vývojářské sady AIR SDK. Tuto širokou dostupnost zajišťuje úzká integrace Flash playeru s WebKit frameworkem (knihovna pro vykreslování HTML a interpretaci JS). V praxi to znamená, že Flash může být klasicky vnořen do HTML struktury nebo naopak vykresluje samotná data umístěná v HTML. Obdobně to platí pro jazyky JavaScript a ActionScript, které se mohou navzájem volat nebo v případě AS manipulovat s HTML DOM.

V neposlední řadě je důležitá integrace části databáze SQLite pro lokální uložení dat a zařazení podpory formátu PDF, pro snadné používání a čtení dokumentů.

Prostředí AIR tak přemostňuje technologie, které se řadu let používaly odděleně a představuje možnost pro webové aplikace osvobodit se od potřeb internetového prohlížeče.



Obrázek 21. Architektura aplikací Apollo (dnes AIR).

Podrobněji v [2], [8]

5.2.6 Budoucnost v maximální integraci

Hlavním důvodem, který by měl společnosti přesvědčit k investování do nástrojů Adobe, je fakt, že jsou schopna zajistit těsnou spolupráci jednotlivých fází vývoje. Dnes nepředstavuje žádný problém naimportovat video z After Effects do Flashe, který pak lze umístit do Flex aplikace. Stejně tak prostřednictvím Flash Playeru nebo AIR mohou být zobrazovány dokumenty PDF. Maximální integrace technologií a formátů, které defacto dnes představují rozšířené standardy, umožňuje rychlou a efektivní práci vedoucí k mnoha tvůrčím řešením.

V této souvislosti by aktuální novinkou mělo být vydání Flash Catalyst, jejímž cílem je ještě více zúžit spolupráci mezi programátorem a grafikem. Catalyst je určen interface designérům, kteří do něj nahrají vytvořené grafické prvky z Photoshopu nebo Illustratoru, jenž jsou definovány v novém grafické formát FXG podporovaným v MXML. Takto vzniklý prototyp uživatelského rozhraní předají programátorovi, který je ve Flexu začlení do dané aplikace.

Produkty společnosti Adobe se staly synonymem pro DTP nebo tvorby grafiky jako takové, avšak s příchodem Flash technologií se její aktivity výrazně přesouvají do oblasti webu. Vytváří tak pro bohaté internetové aplikace myšlenku, kterou snadno spojuje práci lidí napříč jejich profesním spektrem ať se jedná o programátora, tvůrce videa nebo grafika.

Více informací na [15]

5.3 Windows Presentation Foundation

Na fakt, že bohaté grafické rozhraní má svůj nemalý význam pro dnešní software sazí i Microsoft, který tak s příchodem operačního systému Windows Vista představil technologie Windows Presentation Foundation (WPF), které jsou zaměřené na jednotný přístup k uživatelskému rozhraní, dokumentům a médiím. WPF tvoří grafický podsystém frameworku .NET 3.0, kde umožňuje oddělit aplikační logiku od samotného designu. Jeho základní pilířem je další formát XML, jazyk XAML (eXtensible Applications Markup Language), pomocí něhož se definuje vzhled UI. Momentálně je podporován pouze na systémech Windows Vista, Server 2008, XP Service pack 2 a Server 2003, avšak určitou přenositelnost mezi různými platformami představuje technologie Silverlight, která je na základech WPF postavena.

Silverlight se koncepčně velice podobá Flash Playeru. Poprvé se objevil v roce

2007 a o rok později na něj navázala podstatně vylepšená verze 2.0. Opět se jedná o zásuvný modul do webového prohlížeče, který umožňuje přehrávání vektorové grafiky, animací, videa a v neposlední řadě fungování RIA. Microsoft tak s jeho uvedením nepopíratelně dává najevo, že bude mít velké zájmy v této oblasti a vstoupil tak do přímé konkurence s Adobe.

Vzhledem k tomu, že se další tematický celek bude zabývat srovnáním těchto technologií, dle úvahy autora by bylo zbytečné věnovat zvláštní prostor pro výčet detailních charakteristik ohledně funkcionality nebo samostatného vývoje, proto následující text bude spíše zaměřen na zásadní odlišnosti, které tato technologie představuje.

5.3.1 Silverlight

Silverlight principiálně vytváří paralelu pro Flash, respektive Flex, již zmiňovaným jazykem XAML, který deklaruje vizuální stránku prezentační vrstvy, nicméně značnou výhodou je, že tento plugin je postavený na jádru frameworku .NET, který tak umožňuje psát aplikační logiku v mnoha jeho jazycích. Programátoři, kteří tak vytvářeli desktopové aplikace pro .NET mohou na základě svých zkušeností bez problému přejít do oblasti webu. Na výběr mají C# a Visual Basic, ale i další podporované jazyky jako JavaScript Ruby, Python nebo PHP.

Sám o sobě, Silverlight již obsahuje část .NETu , tedy nevyžaduje jeho instalaci na klientském počítači, takže je nezávislý na platformách a prohlížečích. V této souvislosti stojí za zmínku lokalizace pro Linux, kterou zajišťuje společnost Novell jako opensource implementaci s označením Moonlight.“

Společně se Silverlightem vyšla řada produktů, které stejně tak jako v případě Adobe Flash Platform nabízejí nástroje orientované na jednotlivé fáze vývoje, jenž rovněž staví na vzájemně silné integraci.

V přímé souvislosti s RIA je třeba uvést Expression Blend, WYSIWYG editor, který se svou filosofií by se dal přirovnat k nástroji na pomezí Flashe a Flexu, o čemž svědčí jeho zaměření na tvorbu vizuálně atraktivních aplikací - samozřejmě v objektovém modelu jazyka XAML. Podstatným faktem je podpora této technologie i ve Visual Studiu 2008, čímž jí dává velice robustní vývojové prostředí.

Ambice Microsoftu jsou více než jasné a v tisku zaměřeném na IT se začínají skloňovat výrazy jako „Flash Killer“ nebo „RIA války“. Jak tento boj skončí asi ukáže jen čas. Zajímavostí na závěr je, že v březnu, v době kdy autor dokončoval bakalářskou

práci, vyšla testovací verze Silverlight 3.0, která se defacto rozšiřuje o funkce, ve kterých byl Flash Player 10 novinkou (hardwarová akcelerace, 3D, grafické filtry atd.). Nejdůležitějším zásadní zlomem je možnost spustit dané aplikace mimo web, což ruší unikátní postavení rok starého Adobe AIR.

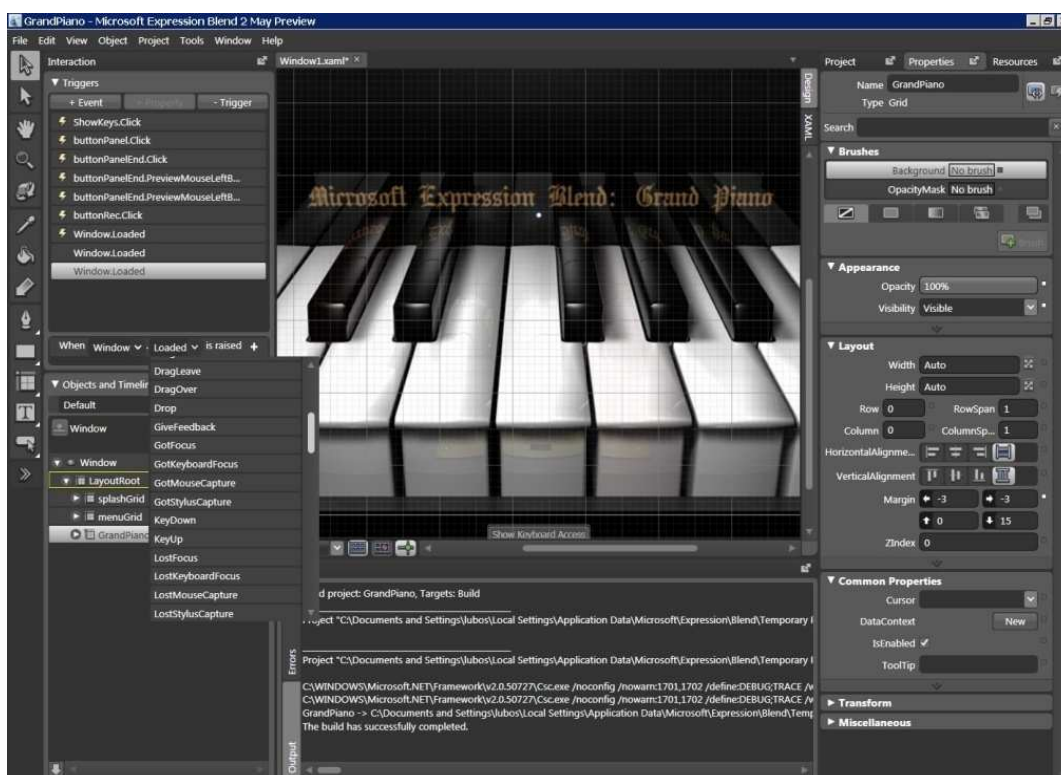
Zpracováno dle dostupných informací na [19], [29], [30]



Obrázek 22. Silverlight logo



Obrázek 23. Moonlight logo



Obrázek 24. Prostředí nástroje Expression Blend

5.4 JavaFX

Ještě před několika lety se o Javě mluvilo jako o mrtvé technologii pro potřeby aplikací na straně klienta. Po neúspěšné éře appletů v 90. letech, však nabírá druhý dech. S rokem 2008 Sun Microsystems představil jednu z nejvýznamnějších inovací této platformy. Nová JavaFX sází na velké rozšíření tohoto jazyka na nejrůznějších počítačových zařízeních a silné vývojářské zázemí.

„Platforma Sun JavaFX 1.0 otevírá obrovský globální trh autorům obsahu a vývojářům, kteří chtějí uplatnit svůj obsah, služby a zkušenosti na všech přístrojích používaných jejich zákazníky. Z odhadů celosvětového průmyslu vyplývá, že technologie Java je už nasazena na více než 90 procentech stolních a přenosných počítačů, 85 procentech mobilních zařízení a je také nejrozšířenější technologií v televizích další generace, přehrávačích Blu-ray disků i televizních set-top boxech.“ [21]

Mezi mnohé inovace bude patřit revoluční funkce Drag-to-Install, která poprvé uživateli umožní přetahovat applety z webového prohlížeče do počítače a tradičně zaměřená vylepšení na integraci grafiky, videa, zvuku a animace. S novou vlnou tak přichází i sada nových vývojových nástrojů a zásuvných modulů JavaFX Development Environment a JavaFX Production Suite, který zajišťuje spolupráci se standardními grafickými nástroji Adobe Photoshop nebo Illustrator.

Pokud se Sun vyvaruje svých chyb z minula, představuje pak dalšího rivala do pomyslných RIA válek, který nebude muset řešit problémy s masovým rozšířením narozdíl od Microsoftu.

Více na [21], [26]

6 Srovnání vybraných technologií

V následujícím textu se autor bakalářské práce pokusí porovnat webové technologie Ajax, Flash Player a Silverlight na základě několika kritérií. Hlavní důraz je kladen na jejich možnosti a využití pro oblast RIA. V každé kategorii, autor subjektivně ohodnotí dané zástupce, v rozmezí 0-5 bodů. Provede tak na základě vlastních zkušeností a dostupných informací.

Platformní nezávislost

V případě nasazení již hotové webové aplikace je velice důležitým aspektem v jaké míře jsou klientské technologie podporovány na jednotlivých platformách. Podpora na

nejpoužívanějších prohlížečích a operačních systémech je zobrazena v tabulce č. 1. Všichni tři zástupci si v tomto ohledu vedou takřka stejně. U hodnocení nebyla brána v potaz rozdílná kvalita některých Ajaxových frameworků, které mají problémy s kompatibilitou v různých prohlížečích. Dále bodování neovlivnila občasná chybovost pluginů, které bývá většinou následně odladěna. Ajax: 5, Flash: 5, Silverlight: 4

	Ajax	Flash Player	Silverlight
Internet Explorer	Ano	Ano	Ano
Firefox	Ano	Ano	Ano
Opera	Ano	Ano	Ne
Safari	Ano	Ano	Ano
Windows Vista	Ano	Ano	Ano
Windows XP	Ano	Ano	Ano
Mac OS X	Ano	Ano	Ano
Linux	Ano	Ano	Ano (verze Moonlight)
Solaris	Ano	Ano	Ne

Tabulka 1. Platformní nezávislost

Rozšířenost

Konkrétně u Ajaxu je rozšířenost brána jako procento uživatelů, kteří nemají ve svém prohlížeči vypnutý JavaScript. Ve vztahu k pluginům se jedná o procento uživatelů, kteří ho mají nainstalovaný na počítači. Ajax: 5, Flash: 5, Silverlight: 2

	Ajax	Flash Player	Silverlight
Rozšířenost	99%	99%	30%

Tabulka 2. Rozšířenost

Programovací jazyky

V tomto ohledu jednoznačně vítězí Silverlight, který nabízí celou škálu jazyků pro .NET. Programátoři Flashových aplikací mohou použít jen ActionScript, eventuelně JavaScript pro framework OpenLaszlo. Stejně tak Ajax staví svou aplikační logiku na JavaScriptu, nicméně existuje řada dalších řešení jako Google Web Toolkit, které umožňují psát i v jiných jazycích.

Ajax: 2, Flash: 2, Silverlight: 5

Výkon

Z hlediska výkonu jsou si obě dvě pluginové technologie takřka rovnocenné. Současnou výhodou přehrávače Flash je částečná hardwarová akcelerace, kterou Silverlight

umožňuje zatím ve své betaverzi. Nejvíce zaostává Ajax se svým JavaScriptem, jehož kód se před nasazením nekompile a běží přímo v prohlížeči.

Ajax: 2, Flash: 5, Silverlight: 4

Dostupnost vývojových nástrojů

V počtu nejvíce dostupných knihoven, aplikačních rámců nebo komponentových aplikačních rámců, jednoznačně vede Ajax. Na druhou stranu Adobe a Microsoft nabízejí mimo svá vývojová prostředí i řadu specializovaných aplikací zaměřených na design. V případě platformy Flash jde o nástroje, které za sebou mají již řadu let vývoje a pro mnoho tvůrců se staly uznávanými standardy. Microsoft se svým balíkem Expression sice směřuje také na tyto pozice, ale bude trvat ještě nějaký čas, než si mezi uživateli najde své místo. Neposlední důležitou stránkou věci je, že většina Ajaxových frameworků je dostupná zdarma, kdežto Flash, Flex, Blend a další jsou komerční produkty.

Ajax: 5, Flash: 4, Silverlight: 2

Vývojářská komunita

Neméně důležitým aspektem je i velikost vývojářské komunity. Pokud se někdo zaměří na vývoj na určité platformě, bezpochyby ocení dostupnost tutorialů nebo už hotových řešení, ze kterých může v případě nějakého problému čerpat. V případě Ajaxu a Flashe se na Internetu najde mnoho serverů, kde se dělí programátoři o své zkušenosti. Ačkoliv Silverlight staví na rozšířeném .NET, stejně představuje novou technologii, která si musí určitou skupinu uživatelů teprve vybudovat.

Ajax: 5, Flash: 5, Silverlight: 2

Animace

Flash i Blend disponují animačními modely na velmi vysoké úrovni. Nicméně Adobe nabízí i řadu dalších vyspělejších nástrojů s tímto zaměřením (například After Effects), které s Flashem mohou spolupracovat. S JavaScriptu lze také provádět animace, ale je to daleko složitější než za pomoci specializovaného softwaru.

Ajax: 2, Flash: 5, Silverlight: 4

Media

Zprvopočátku měl nový Silverlight značnou výhodu oproti Flash Playeru 9, mezi kterou patřilo možnost přehrávání videa v kvalitě HD v režimu přes celou obrazovku. Verze 10 tyto nedostatky odstranila, přesto stále pro svá videa využívá méně rozšířený formát FLV, na rozdíl od svého konkurenta, který přehrává velice používaný WMV. Mimo jiné Microsoft každému nabízí 4GB volného prostoru pro uložení dat na svém streamovacím

serveru. V Ajaxu není jakákoliv možnost přehrávání, proto je odkázán na použití softwaru třetích stran.

Ajax:0, Flash:3, Silverlight: 5

Dle výsledné sumarizace bodů, která je uvedené v tabulce č.3 se jako nejlepší jeví Flash. Na jeho hodnocení mělo největší vliv to, že nabízí robustní vývojové prostředí a je poměrně rozšířenou technologií.

Nízký počet bodů u Ajaxu, hlavně zapříčinila jeho špatná práce s grafikou a nulová podpora medií, přestože Silverligh nabízí velice podobné možnosti jak Flash dopadl téměř stejně jako Ajax, díky tomu, že je poměrně mladou technologií, která ještě není takovým způsobem rozšířená tak jako jeho konkurenti.

	Ajax	Flash Player	Silverlight
Platformní nezávislost	5	5	4
Rozšířenost	5	5	2
Programovací jazyky	2	2	5
Výkon	2	5	4
Dostupnost vývojových nástrojů	5	4	2
Vývojářská komunita	5	5	2
Animace	2	5	4
Media	0	3	5
Celkem	26	34	28

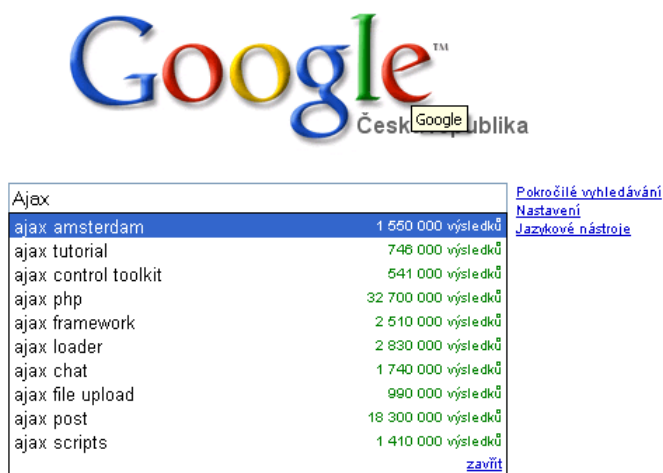
Tabulka 3. Sumarizace

Jednotlivé parametry dostupné na [21], [25], [31], [32]

7 Implementace RIA – příklad řešení

Příkladů využití výše probraných technologií je na Internetu k nalezení celá řada. Co se týče Ajaxu často se využívá na webových stránkách k simulacím funkce drag& drop, vytváření oken s nápovědou nebo k nejrůznějším našeptávačům. Stejně tak s Flashem se nejčastěji setkáme v podobě reklamních bannerů, on-line her nebo dynamických součástí klasických HTML stránek, tzv. hybridní weby. Proto je třeba si uvědomit, že pojem Ajax, Silverlight nebo Flash se zákonitě nerovná RIA.

Na druhou stranu, bohatých webových aplikací, které se komplexně opírají pouze o jednu z uvedených technologií, je dnes již poměrně velký počet, proto následující kapitola bude patřit pouze jim.



Ajax		Pokročilá vyhledávání
		Nastavení
		Jazykové nástroje
ajax amsterdam	1 550 000 výsledků	
ajax tutorial	746 000 výsledků	
ajax control toolkit	541 000 výsledků	
ajax php	32 700 000 výsledků	
ajax framework	2 510 000 výsledků	
ajax loader	2 830 000 výsledků	
ajax chat	1 740 000 výsledků	
ajax file upload	990 000 výsledků	
ajax post	18 300 000 výsledků	
ajax scripts	1 410 000 výsledků	
		zavřít

Obrázek 25. Našeptávač vyhledávače Google



Obrázek 26. Ukázka hybridní stránky, horní lišta a navigace běží v přehrávači Flash

7.1 Google Maps, Mail, Docs...

Asi za jednoho z největších propagátorů Ajaxu by se dala považovat společnost Google, která ho používá pro vývoj mnoha svých služeb. Jedny z prvních RIA, které Google představil byl webový poštovní klient Gmail a systém světových map Google Maps, které netřeba více charakterizovat, protože je dnes využívá mnoho lidí.

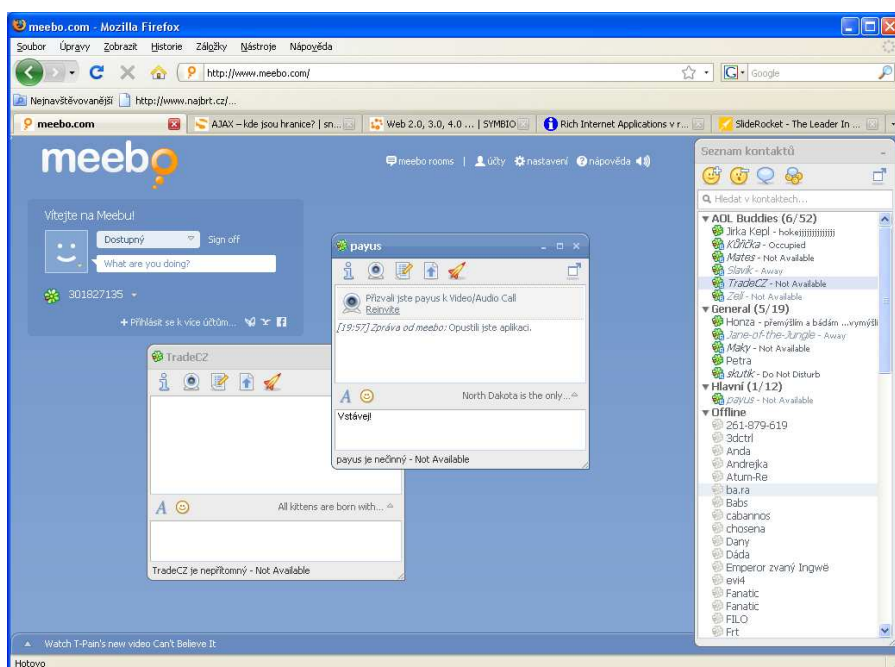
Neméně známá je služba Google Docs, která představuje alternativu ke klasickým kancelářským balíčkům jako MS Office nebo Open Office. Přestože splňuje základní funkcionalitu svých desktopových protějšků, neznamena pro ně momentální hrozbu. Nezávislá studie uvádí, že podíl využití Google Docs na americkém trhu je okolo 1%. Základní příčinou tohoto neúspěchu nepředstavují technologická omezení, ale konzervatismus současných uživatelů. [33]

Název činnosti	Ce	á práce (h)	Zdroj (intenzita čerpání)	Předchází	Normy pře
Projekt Výstavba domu					
Projektové práce	30d		Projektant		
Projektování	odví	120	Zedníci[4]	Projektování	140
Zahajovací fáze	odví	a 300	Zedníci[2], Řemeslníci[2]	Úprava terénu (SS+14d)	27,5 a 37,5
Úprava terénu	odvozeno	3840	Zedníci[8]	Připojky	480
Připojky					
Základy			Písek (35 m3), Cihly 4500 ks, Cement 10t		0
Stavba					
Stěny	odvozeno	7300	Zedníci[8]	Základy	912,5
Střecha	odvozeno	1280 a 400	Tesaři[4], Zedníci[2]	Stěny	160 a 80
Instalace	odvozeno	1300 a 200	Řemeslníci[6], Zedníci[3]	Stěny	162,5 a 25
Vnitřní omítky	odvozeno	600	Zedníci[4]	Instalace (SS+20d)	75
Garáž	odvozeno	48 a 600	Tesaři[1], Zedníci[3]	Stěny	6 a 75
Vnější omítky	odvozeno	1000	Zedníci[6]	Střecha, vnitřní omítky	125
Závěr					
Úklid	10d	odvozeno	Zedníci[4]	Vnitřní a vnější omítky	
Kolaudace	1d	odvozeno	Projektant	Úklid	
Projekt začíná 1.4.2009 a není-li uvedeno jinak					
Pracuje se Po-Pá 8:00 až 16:00 bez polední					
Činnost "Stěny" musí začít nejdříve 1.3.201					

Obrázek 27. Tabulkový procesor ze sady Google Docs.

7.2 Meebo.com

Na Ajaxu je také postavená aplikace Meboo. Služba umožňuje přístup k řadě instant messaging sítím a díky rozsáhlé podpoře těchto protokolů nemá ve své kategorii konkurenci. Prostřednictvím webového prohlížeče se uživatelé odkudkoliv dostanou na ICQ, MSN, MySpace, Google talk nebo Facebook chat. Pokud někdo vlastní více účtů na různých službách může je v rámci Meeba spojit do jednoho. Uživatelské rozhraní napodobuje chování oken klasického ICQ klienta na ploše operačního systému. Ačkoliv jeho základní myšlenkou je přistupovat k uživatelům z webu, nedávno se objevila i desktopová verze Meebone běžící na Adobe AIR.

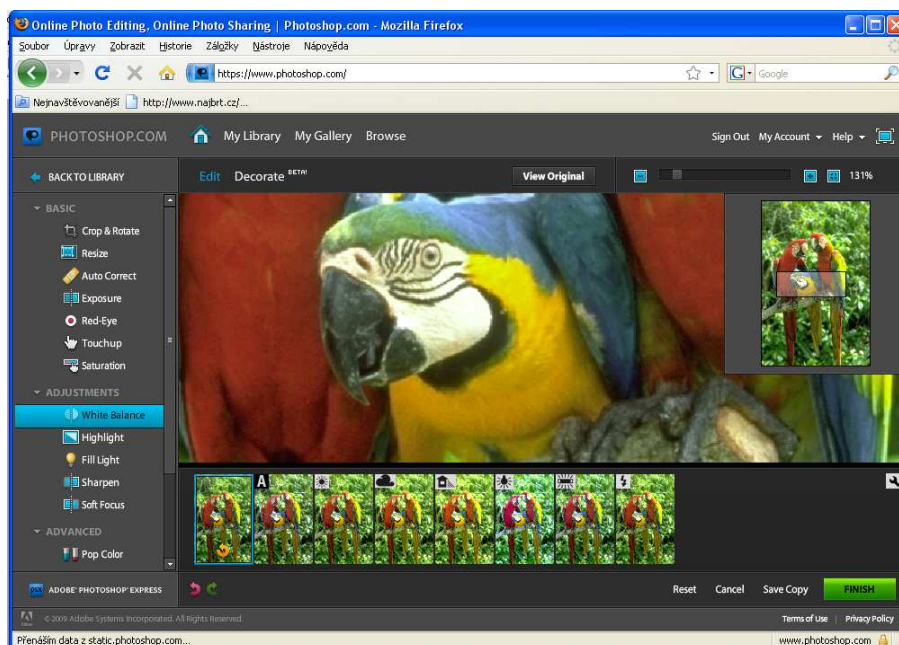


Obrázek 28. ICQ klient v Meebo

7.3 Photoshop.com

Na platformě Flash Adobe vystavěla on-line verzi svého Photoshopu jako reakci na stále silnější popularitu programů, které umožňují úpravu fotografií v prostředí webu. Na rozdíl od své desktopové verze nabízí jen zlomek základních funkcí pro úpravu barevnosti, editaci rozměru nebo vkládání textu. Každý registrovaný uživatel má k dispozici 2 GB prostoru pro svá alba, která může sdílet s ostatními uživateli. Zajímavá je integrace s dalšími webovými galeriemi jako jsou Flickr, Picasa nebo Facebook, odkud lze fotografie nahrát a upravené vrátit zpět. V této návaznosti je třeba

se zmínit, že Adobe nabízí i další podobné službě jako je Acrobat.com pro zprávu a vytváření dokumentu PDF nebo textový procesor Buzzword. [33]



Obrázek 29. Photohop.com

7.4 NASDAQ Market Replay

Největší elektronický trh s akciemi NASDAQ se přiklonil k Adobe Flash Platform a na AIR klientovi vybudoval svou obchodní aplikaci NASDAQ Market Replay. Každý, kdo má zájem jeho služeb využít, si je musí nainstalovat na svůj počítač. Aplikace poté z Internetu získává a ukládá data spojená s vývojem na trhu akcií. Klíčovou funkcí je možnost si celý proces vývoje zpětně přehrát, jako by probíhal v reálném čase. [22]



Obrázek 30. NASDAQ Market Replay

7.5 Oracle CRM Gadgets

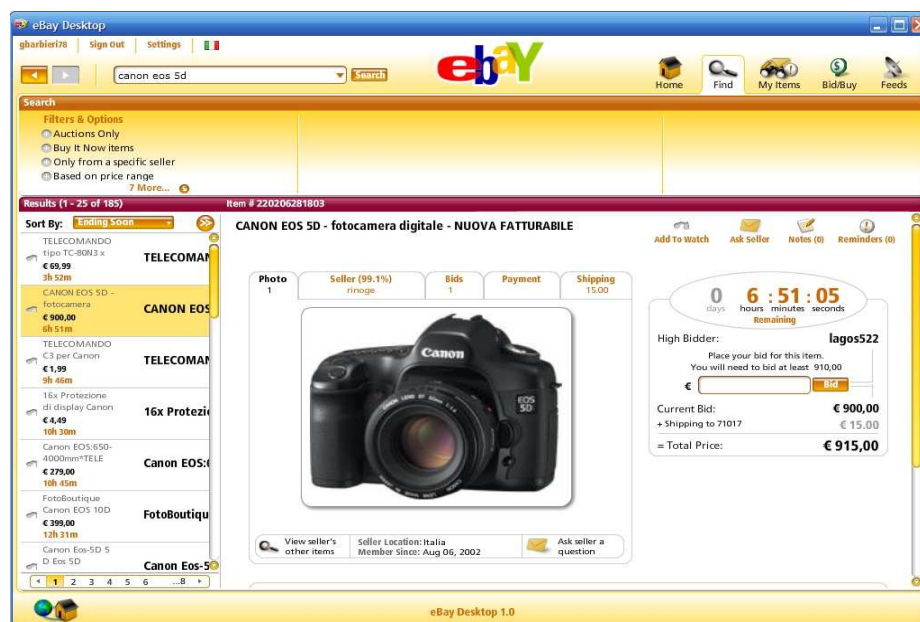
Myšlenku desktopových widgetů a gadgetů uchočila i společnost Oracle Corporation, která využila možnosti vyvíjet AIR aplikace pomocí HTML a JavaScriptu. Na této bázi vytvořila rodinu miniaplikací Oracle CRM Gadgets, které spolupracují se CRM systémem Siebel a rovněž získávají data z Internetu. Gadgety zobrazují kontakty na důležité zákazníky, přehledy jednotlivých prodejů a nejrůznějších kvót atd. [27]



Obrázek 31. Oracle CRM Gadgets.

7.6 eBay Desktop

Trojici AIR aplikací uzavírá eBay Desktop, který je určen častým návštěvníkům stejnojmenné elektronické dražby. Uživatelé umožňuje odprostit se od služeb webového prohlížeče a být trvale přihlášen ke svému osobnímu účtu. Hlavní přednost spočívá v eliminaci přenesených dat, která způsobuje daleko rychlejší odezvu ze strany uživatelského rozhraní aplikace, což je neocenitelnou výhodou při uzávěrkách jednotlivých dražeb, kdy o úspěchu rozhodují pouhé vteřiny.



Obrázek 32. eBay Desktop

7.7 G.ho.st

Další aplikací využívající Flash Player představuje poněkud odvážný projekt G.ho.st. Odvážný v tom ohledu, že stejnojmenná firma nabízí prostřednictvím webu přístup na vlastní virtuální operační systém. Tentokrátě vývojářům usnadnil práci framework Open Laszlo a celkový vzhled uživatelské rozhraní je výrazně stylizován do MS Windows. Stejně tak jako klasický operační systém na stolním počítači, G.ho.st. může nabídnout emailového klienta, webový prohlížeč, audio přehrávač, textový editor nebo souborový systém s diskovým prostorem 5GB. Ačkoliv se jedná o aplikace, se kterými se uživatel setkává každodenně, v současnosti služby tohoto virtuálního stroje, vážně využije asi jen málokdo. Nicméně do budoucna webový operační systému zůstává velice atraktivní myšlenkou, která však svým rozměrem sahá hluboko za hranice současných RIA.

Více informací dostupných na [34], [35]



Obrázek 33. Virtuální OS G.ho.st

7.8 NBColympics.com

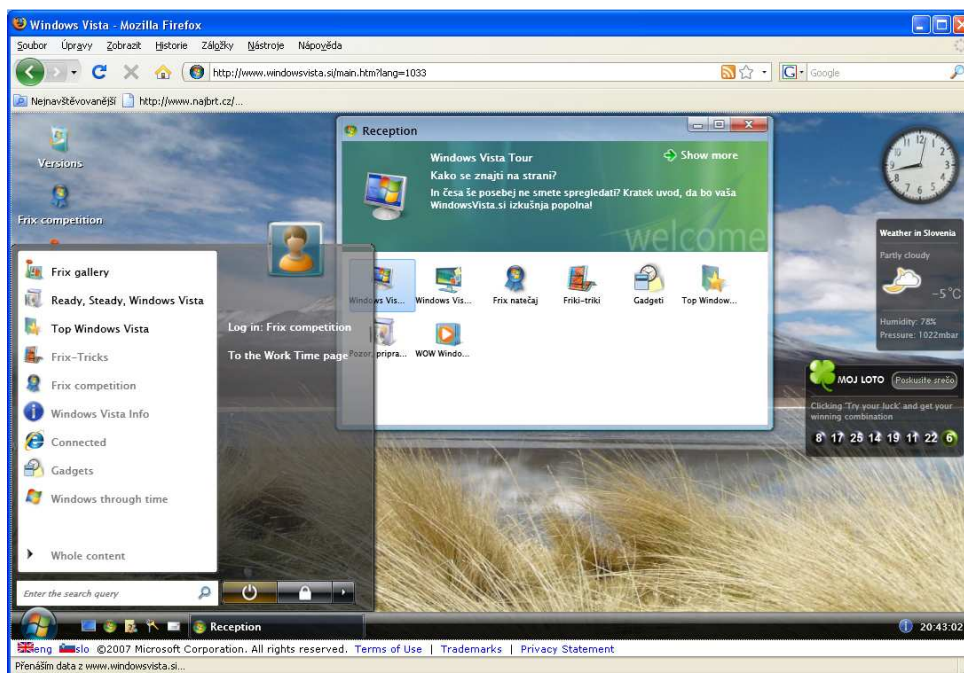
Ačkoliv má Silverlight na svém webu poměrně bohatý showcase, kde vývojáři z celého světa mohou vystavovat své aplikace, dle autorova shlednutí na pořádnou Rich Internet application stále ještě čeká. Na ukázkách je patrné, že je většinou využívána k dílčím řešením, nebo jako prostředí pro interaktivní webové prezentace. Historicky doposud k největšímu nasazení Silverlightu došlo na serveru televizní stanice NBC, která pod doménou nbcolympic.com nabízela zpravodajství z letních Olympijských her v Pekingu, a Silverlight zde posloužil jako prostředí pro přehrávání sportovních přenosů. Díky tomu se tak během krátkého období rozšířil na mnoho počítačů sportovních fanoušků. [24]



Obrázek 34. Silverlightové aplikace na NBColympics.com

7.9 Windowsvista.si

V prostředí Silverlightu na slovinské doméně www.windowsvista.si běží ukázková maketa stejnojmenného operačního systému. Vzhledem k tomu, že se omezuje pouze na prezentaci chování a vizuální složky uživatelského rozhraní, a z funkcionality Vist nenabízí téměř nic, balancuje tak na pomezí mezi efektní webovou prezentací a Rich Internet Application. [19]



Obrázek 35. Simulace OS Windows Vista.

8 Očekávaný vývoj

Interaktivní webové aplikace jsou na trhu stále žádanější, a tak na scénu s technologiemi vstoupilo několik velkých softwarových firem, které přinášejí poměrně originální řešení. Nabízí se paralela s HTML, kdy v konkurenčním boji prohlížečů v 90. letech doznalo silné nekonzistence, což v důsledku poškodilo hlavně koncového uživatele. Stejná situace může nastat i v případě velkého množství proprietárních pluginů, které řadového návštěvníka webu budou neustále zatěžovat instalací nebo upgradem další verze. V tomto ohledu silný konkurenční boj situaci jedině prospěje, a dá se jen doufat, že se ustálí na dvou, třech majoritních zástupcích.

Důležitým problémem je samotná role webových prohlížečů, na kterých je přímo závislý pouze Ajax, zatímco pro pluginové technologie je spíše překážkou. Do budoucna se očekává nárůst RIA, které své klienty budou instalovat na desktop, tak jako se děje v případě Adobe AIR a vyšlé betaverze Silverlightu 3.0. Stejně tak se objevují názory, že končí éra zmiňovaného Ajaxu, který kromě asynchronní komunikace se v jiných ohledech svým konkurentům nemůže rovnat.

Ajax je excelentní trik, který udává poslední cestu JavaScriptu, je to však trik neúspěšný, který má svůj konec v dohledu.[1]

Nejvíce diskutovanou otázkou, je zdali v blízké budoucnosti, webové aplikace nenahradí ty klasické na stolních počítačích. Autor je toho názoru, že ne. Ať už je to Photosop.com nebo Google Docs, stále nabízejí jen malou funkcionalitu oproti svým desktopovým konkurentům. Velkou roli v tom sehraje dostupnost kvalitního připojení k Internetu a důvěra pro web jako aplikační platformu.

RIA jsou teprve na svém začátku a jejich momentální orientace není bojovat s monolitickými programy, ale směřuje k tomu, být zkrátka „všude.“ Důkazem toho je iniciativa fondu Open Screen Project, jenž společně řídí Adobe a Nokia. Jeho smyslem je vytvořit konzistentní provozní prostředí pro webové aplikace postavených na Adobe Flash platformě, které budou pracovat na mobilních telefonech, stolních počítačích, televizorech a dalších přístrojích spotřební elektroniky.

Hlavní oblasti, které si fond dává za úkol, jsou: zábava, sociální sítě, produktivita, hraní, zprávy a informace .[36]

9 Závěr

Koncept Rich Internet Application přinesl řešení jak docílit kvalitního a bohatého uživatelské rozhraní webových aplikací, stejně tak jako mají aplikace na desktopu.

To, že představují světlou budoucnost, dokazují aktivity mnoha softwarových společností, která pro tuto oblast v silném konkurenčním boji intenzivně vyvíjí vlastní technologická řešení.

Co se týče samotného vývoje, největší progresy představuje nástup tzv. komponentových aplikačních rámců, které programátorům usnadňují a urychlují práci. Vzhledem k tomu, že v dnešní době grafické uživatelské rozhraní musí splňovat i estetickou stránku, vývojové nástroje se neobejdou bez silné integrace s ostatními programy pro vytváření grafiky a multimédií.

Mnoho RIA je nezávislých na možnostech, které jim poskytují webové prohlížeče, protože pracují ve vlastním běhovém prostředí (runtime) na straně klienta. Nejčastěji jsou řešeny jako zásuvné moduly (plugin) do internetového prohlížeče, avšak přicházejí na řadu runtimy, které fungují rovnou na desktopu a programy, kterým umožňují běh, využívají web jen pro stahování aktualizací dat. Paradoxně se tak RIA přesouvají mimo web a do budoucna bude snahou vytvořit co nejširší podporu napříč jednotlivými platformami, které umožní spustit stejné aplikace nejen na osobním počítači nebo mobilním telefonu, ale i na další řadě domácích elektrospotřebičů.

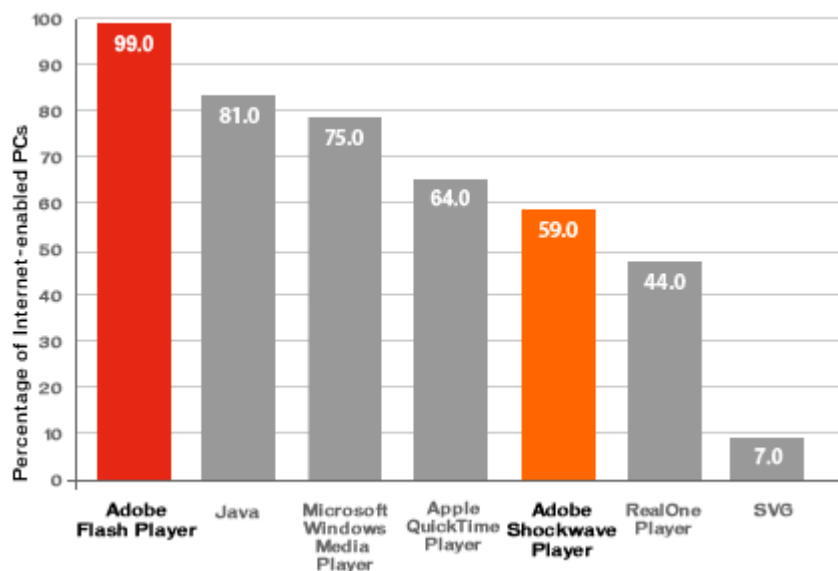
10 Seznam literatury

1. FAIN, Y.; RASPUTNIS, V.; TARTAKOVSKY, A.: Rich Internet Applications with Adobe® Flex™ & Java™ Secrets of the Masters. USA, SYS-CON Books, 2007. 736 s.
ISBN 0-9777622-2-X
2. CHAMBERS, M.; DIXON, R.; SWARTZ, J.: Adobe Apollo® for Flex™: Pocket Guide. USA, O'Reilly Media Inc., 2007. 108 s.
ISBN: 0-596-51391-7
3. SWANN, C.; CAINES, G.: Flash s využitím XML - tvorba dynamických webových stránek. Praha, Grada Publishing a.s., 2003. 240 s.
ISBN: 80-247-0389-0
4. YANK, K.; ADAMS, C.: Začínáme JavaScript. Brno, Zonerpress, 2008. 320 s.
ISBN: 978-80-86815-94-7
5. ASLESON, R.; SCHUTTA, S.: Ajax - Vytváříme vysoce interaktivní webové aplikace. Brno, Computer Press, 2006. 269 s.
ISBN: 80-251-1285-3
6. JACOBS, S.: Beginning XML with DOM and Ajax. USA, Apress, 2006. 390 s.
ISBN: 978-1-59059-676-0
7. TUCK, M.: The Real History of the GUI. [online].
<<http://www.sitepoint.com/article/real-history-gui>>
8. HOLZSCHAG, M.: HTML a CSS jdi do toho. Praha, Grada, 2006. 257 s.
ISBN: 80-247-15145-X
9. SEKERA, J.: Václav Krejčí – design grafiky GUI. [online].
<<http://www.designportal.cz/rozhovory/rozhovor-vaclav-krejci.html>>
10. PICHLÍK, R.: Rich Internet Application. [online].
< <http://interval.cz/clanky/rich-internet-application>>.
11. BERNARD, B.: Rich Internet Applications v roce 2008. [online].
< <http://interval.cz/clanky/rich-internet-applications-v-roce-2008>>.
12. VESELKA, A.: Obohat'te své uživatele pomocí RIA. [online].
<<http://www.symbio.cz/clanky/obohatte-sve-uzivatele-pomoci-ria.html>>

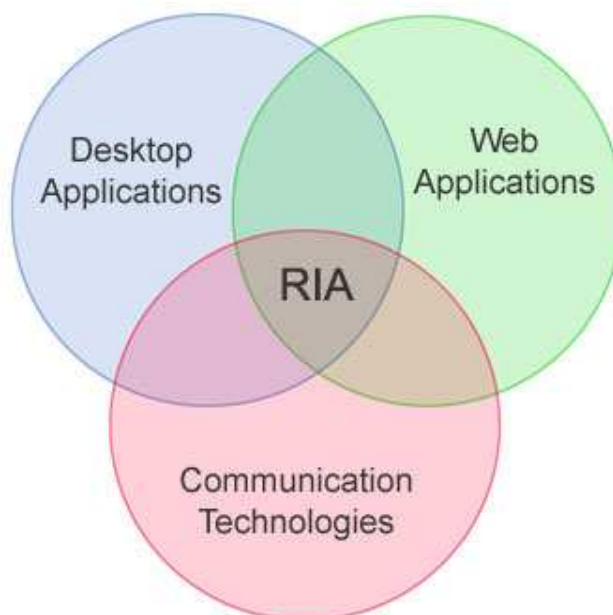
13. ALLAIRE, J.: Macromedia Flash MX—A next-generation rich client. [online].
<<http://download.macromedia.com/pub/flash/whitepapers/richclient.pdf>>
14. BERNARD, B.: Flex co je a co není. [online].
<<http://interval.cz/clanky/adobe-flex-co-je-a-co-neni>>
15. MULLER, A.: Building RIAs with the Adobe Flash Platform. [online].
<<http://www.sitepoint.com/article/rias-adobe-flash-platform-3>>
16. RAZA, I.: Adobe Flex. [online].
<<http://imamrazaansari.blogspot.com/2008/11/adobe-flex.html>>
17. FOJTL, J.: K čemu slouží Adobe Flex. [online].
<<http://zdrojak.root.cz/clanky/k-cemu-slouzi-adobe-flex>>
18. ČÍŽEK, J.: Adobe AIR Beta 3: WEB 2.0 přichází na desktop. [online].
<<http://www.zive.cz/Clanky/Adobe-AIR-Beta-3-WEB-20-prichazi-na-desktop/sc-3-a-139665/default.aspx>>
19. LACKO, L.: Úvod do technológie a tvorby aplikácií pomocou nástrojov Microsoft Expression. [online].
<http://download.microsoft.com/download/0/a/6/0a633751-793c-49cc-be8b-bf2f283e9f74/Silverlight_Expression_Slovak.pdf>
20. BURANSKÝ, I.: XML a webové služby. [online].
<http://download.microsoft.com/download/8/6/c/86c09926-affc-4e14-bec0-3c45cd989436/XML_a_webove_sluzby.pdf>
21. RYVOLA, P.: Sun Microsystems uvádí platformu JavaFX 1.0 s bohatými multimediálními funkcemi pro 800 milionů počítačů s Javou na trhu. [online].
<http://www.virklis.cz/cs/press/sun-microsystems/javafx1/_files/0812-sun-uvadi-platformu-javafx-1-0-s-bohatymi-multimedialnimi-funkcemi.doc>
22. Adobe Systems. [online].
< <http://www.adobe.com> >.
23. Laszlo Systems. [online].
<<http://openlaszlo.org>>
24. Microsoft Silverlight. [online].
<<http://silverlight.net>>

25. Microsoft. [online].
<<http://www.microsoft.com>>
26. JavaFX. [online].
<<http://javafx.com>>
27. Oracle. [online].
<<http://www.oracle.com/index.html>>
28. Grafika On-Line. [online].
<<http://www.grafika.cz>>
29. Interval.cz. [online].
<<http://interval.cz>>
30. Zdroják. [online].
<<http://zdrojak.root.cz>>
31. Navrcholu.cz. [online].
<<http://navrcholu.cz/>>
32. Symbio Internetová agentura. [online].
<<http://www.symbio.cz/slovník.html>>
33. NOSKA, M.: Microsoft Office, OpenOffice.org a Google Docs – jak jsou oblíbené? [online].
<<http://computerworld.cz/software/microsoft-office-openoffice-org-a-google-docs-jak-jsou-oblibene-238>>
34. Lupa.cz. [online].
<<http://www.lupa.cz>>
35. Živě Computer. [online].
<<http://www.zive.cz>>
36. Open Screen Project. [online].
<<http://www.openscreenproject.org>>

11 Přílohy



Graf 1. Rozšířenost jednotlivých platforem



Obrázek 36. RIA znázorněná průnikem tří množin

12 Seznam obrázků

Obrázek 1. Operační systém počítače Alto(1973)	6
Zdroj: < http://arstechnica.com >	
Obrázek 2. OS Windows Vista (2007).....	6
Zdroj: < http://www.iconized.ch >	
Obrázek 3. Vztahy mezi značkovacími jazyky.....	9
Zdroj: < cscie153.dce.harvard.edu >, Upraveno autorem	
Obrázek 4. Současné pole působnosti RIA	15
Zdroj: [11]	
Obrázek 5. Modelový elektronického obchodu, který pracuje v prostředí Flash Playeru.	17
Zdroj: Archiv autora	
Obrázek 6. Množiny jednotlivých technologií.....	18
Zdroj: [11]	
Obrázek 7. Strom jednotlivých elementů DOM	19
Zdroj: < http://javascript.about.com >	
Obrázek 8. Klasický a ajaxový webový aplikační model	20
Zdroj: [10]	
Obrázek 9. Normální a ajaxový požadavek stránky.....	21
Zdroj: [4], Upraveno autorem	
Obrázek 11. Asynchronní komunikační model.....	21
Zdroj: < http://www.developer.com >	
Obrázek 10. Synchronní komunikační model	21
Zdroj: < http://www.developer.com >	
Obrázek 12. Ukázka několika komponent jednotlivých aplikačních rámcu.....	24
Zdroj: Archiv autora	
Obrázek 13. Dashboard Widgety na Mac OS X	25
Zdroj: < http://www.math.harvard.edu >	
Obrázek 14. Nástroje a technologi platformy Adobe Flash.....	26
Zdroj: [7], Upraveno autorem	
Obrázek 15. Aplikace Adobe Flash	29
Zdroj: < www.gradware.com >	
Obrázek 16. MXML a ActionSript se zkompilují do souboru SWF.....	30
Zdroj: [16]	
Obrázek 17. Ukázka zdrojového kódu Flex aplikace.....	31
Zdroj: [16]	
Obrázek 18. Flex komponenty	31
Zdroj: Archiv autora	
Obrázek 20. Architektura Laszlo aplikací.....	32

Zdroj: [23]	
Obrázek 19. Architektura Flex aplikací	32
Zdroj: [16]	
Obrázek 21. Architektura aplikací Apollo(dnes AIR).	33
Zdroj: [18]	
Obrázek 23. Moonlightt logo	36
Zdroj: < www.oreillynet.com >	
Obrázek 22. Silverlight logo	36
Zdroj: < http://www.phonesreview.co.uk >	
Obrázek 24. Prostředí nástroje Expression Blend	36
Zdroj: [28]	
Obrázek 25. Našeptávač vyhledávače Google	41
Zdroj: Archiv autora	
Obrázek 26. Ukázka hybridní stránky, horní lišta a navigace běží v přehrávači Flash ..	41
Zdroj: Archiv autora	
Obrázek 27. Tabulkový procesor ze sady Google Docs.	42
Zdroj: Archiv autora	
Obrázek 28. ICQ klient v Meebo	43
Zdroj: Archiv autora	
Obrázek 29. Photohop.com	44
Zdroj: Archiv autora	
Obrázek 30. NASDAQ Market Replay.....	44
Zdroj: < http://www.infoq.com >	
Obrázek 31. Oracle CRM Gadgets.	45
Zdroj: < http://www.zdnet.com >	
Obrázek 32. eBay Desktop.....	45
Zdroj: < http://www.giovy.it >	
Obrázek 33. Virtuální OS G.ho.st	46
Zdroj: Archiv autora	
Obrázek 34. Silverlightové aplikace na NBColympics.com.....	47
Zdroj: < http://blogs.msdn.com/webnext >, Upraveno autorem	
Obrázek 35. Simulace OS Windows Vista.	48
Zdroj: Archiv autora	
Obrázek 36. RIA znázorněná průnikem tří množin.	54
Zdroj: < www.simonwhatley.co.uk >	
Obrázek 37. Možnosti technologie Silverlight.	55
Zdroj: [24]	

13 Seznam tabulek a grafů

Tabulka 1. Platformní nezávislost.....	38
Zdroj: Archiv autora	
Tabulka 2. Rozšířenost.....	38
Zdroj: Archiv autora	
Tabulka 3. Sumarizace	40
Zdroj: Archiv autora	
Graf 1. Rozšířenost jednotlivých platforem.....	54
Zdroj: [22]	