Prezentační rozhraní pro demonstrační PC

Bc. Jaromír Kučerňák

Diplomová práce
2008

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
Ústav aplikované informatiky
akademický rok: 2007/2008

ZADÁNÍ DIPLOMOVÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení:	**Bc. Jaromír KUČERŇÁK**
Studijní program:	**N 3902 Inženýrská informatika**
Studijní obor:	**Informační technologie**
Téma práce:	**Prezentační rozhraní pro demonstrační PC**

Zásady pro vypracování:

1. **Prostudujte problematiku informačních kiosků (prezentačních strojů). Vypracujte přehled jednotlivých technologií vhodných pro prezentační účely.**
2. **Jednu z technologií zvolte pro tvorbu rozhraní za účelem poskytování informací o hardwaru.**
3. **Rozhraní by mělo umožňovat zobrazování WWW stránek, flash animací a spouštění externího programu.**
4. **Celý systém by měl být zabezpečen tak, aby se uživatel nedostal do administrační části systému.**
5. **Vytvořené rozhraní se bude ovládat přes dotykový displej.**

Rozsah práce:

Rozsah příloh:

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

Seznam odborné literatury:

1. **FRANKLIN, Derek. Macromedia Flash MX : Kompletní průvodce. 1. vyd. Brno : Computer Press, 2003. 846 s., 1 CD-ROM. ISBN 80-7226-831-7.**
2. **KERMAN, Phillip. ActionScript ve Flashi : Podrobná příručka. 1. vyd. Praha : Computer Press, 2002. 507 s. ISBN 80-7226-615-2.**
3. **Digital Media s.r.o.. Flash.cz - server pro kreativní lidi : Server o programech Flash, Dreamweaver, Fireworks, Photoshop [online]. c2005-2007 [cit. 2008-01-24]. Dostupný z WWW: http://www.flash.cz/.**
4. **Adobe Systems Incorporated.. Produkty Adobe [online]. c2008 [cit. 2008-01-24]. Dostupný z WWW: http://www.adobe.com/cz/.**
5. **Microsoft Corporation.. Microsoft Česká republika [online]. c2007 [cit. 2008-01-24]. Dostupný z WWW: http://www.microsoft.com/cs/cz/.**
6. **Adobe Systems Inc.. Flex.org : Rich Internet Applications, Rapid Web Application Development, Open Source Flex, Open Source Flash, Adobe Flex, Flex 2, Flex 3 [online]. [2008] [cit. 2008-01-24]. Dostupný z WWW: http://flex.org/.**

Vedoucí diplomové práce: **Ing. Martin Sysel, Ph.D.**
Ústav aplikované informatiky

Datum zadání diplomové práce: **20. února 2008**

Termín odevzdání diplomové práce: **19. května 2008**

Ve Zlíně dne 20. února 2008

prof. Ing. Vladimír Vašek, CSc.
děkan

doc. Ing. Ivan Zelinka, Ph.D.
ředitel ústavu

ABSTRAKT

Cílem práce je vytvořit rozhraní, které zobrazuje informace o hardwarových technologiích. Teoretická část práce je zaměřena na využití informačních kiosků, vysvětluje princip funkce jednotlivých dotykových technologií umožňujících ovládání kiosků skrze displej. V další části je zmíněn princip bohatých internetových aplikací (RIA), jsou popsány významné technologie tvorby RIA aplikací, včetně jejich výhod a nevýhod. Závěr teoretické části se věnuje základům práce s kaskádovými styly a úvodu do programování v jazyku XML.

Praktická část je vedle tvorby rozhraní v prostředí Adobe Flex také zaměřena na problematiku zabezpečení operačního systému Microsoft Windows XP Professional a internetového prohlížeče Opera spuštěného v kioskovém módu.

Klíčová slova: informační kiosek, dotykový displej, RIA, XML, CSS, internetový prohlížeč, Adobe Flex, Windows XP Professional, zabezpečení

ABSTRACT

The aim of this work is to create interface for displaying information concerning hardware technologies. The theoretical part is focused at usage of informative kiosks, explains functions of touch screen technologies, which enable kiosk control through screen. Principle of Rich internet applications including some significant advantages and disadvantages is mentioned in next chapter. The basics of CSS and XML programming are described at the end of this part.

Creation of interface in Adobe Flex interface is the main purpose of the practical part. Another part of this dissertation is to secure operating system Microsoft Windows XP Professional and web browser Opera, which is running in kiosk mode.

Keywords: informative kiosk, touch screen, rich internet application, XML, CSS, web browser, Adobe Flex, Windows XP Professional, system security

Moje poděkování patří vedoucímu bakalářské práce Ing. Martinu Syslovi, Ph.D. za odborné vedení, cenné rady a připomínky i čas, který mi věnoval.

Motto:

" Každý, s kým se v životě potkám, mě v něčem předstihuje. Tak se od něho učím."

Ralph Waldo Emerson

Prohlašuji, že jsem diplomovou práci zpracoval samostatně a použitou literaturu jsem citoval.

Ve Slavičíně 19. května 2008

OBSAH

ÚVOD

Žijeme v době digitálního věku a pryč jsou doby, kdy bylo nutné navštívit knihovnu, pokud jsme chtěli získat specifické informace, nebo být spojeni s okolním světem pomocí telefonu vázaného na přítomnost kabelu směřujícího do ústředny. V současné době nám nic nebrání nadále používat tyto "zastaralé" způsoby získávání informací a komunikace se světem, ale mnohem pohodlnější je sednout si k počítači připojenému k internetu, nebo mít u sebe mobilní telefon a téměř odkudkoliv zavolat na druhou stranu planety. To, že lidé čím dál více upřednostňují "pohodlí" (lépe řečeno komfort), dokazují i statistiky.

Větší část populace má možnost přístupu k internetu ať už v práci nebo v domácnostech. Mimo tyto lokality je přístup k internetu značně omezen. Proto postupně začínají větší města, ale i ty menší zřizovat veřejné přístupové body poskytující informace, tzv. informační kiosky.

Informační kiosek je jednoúčelový stroj, jehož základ tvoří klasické PC. Kiosky se používají nejen jako přístupová místa k internetu nebo informační centra, ale jsou také aplikovány v místech, která jsou dnes nedílnou součástí lidského života. Takovou aplikací jsou například bankomaty, informační systémy letišť, hotelů a nádraží, herní zařízení, výukové trenažéry a v neposlední řadě také automaty na jízdenky a parkovací lístky. Hlavní devízou kiosků je jejich snadné ovládání a to i pro méně technicky zdatné uživatele. Velkou mírou se na snadném používání podílí dotykové technologie, které umožňují ovládat kiosek pohybem prstu po obrazovce. Mezi hlavní technologie dotykových obrazovek patří rezistivní, kapacitní, infračervená a akustická technologie. Každá z těchto technologií se uplatní v jiné oblasti, kde se naplno projeví její výhody. K umístění kiosků na veřejnosti se váže jedna podstatná skutečnost – takové stroje je nutné chránit před vandalstvím. Vedle povrchové ochrany, mezi které patří bezpečnostní skla na displeji nebo kovový kryt, je potřeba ochránit programové vybavení a tím funkčnost kiosku. Tato ochrana se provádí zabezpečením operačního systému, případně internetového prohlížeče.

Cílem této práce je vytvořit rozhraní pro informační kiosek, které bude zobrazovat informace o hardwarových technologiích. Dále pak kiosek náležitě zabezpečit, aby se do systému nedostala nepovolaná osoba, např. za účelem poškodit jeho programové vybavení. Tento informační panel může posloužit studentům informatiky k dalšímu vzdělávání se v hardwarových technologiích.

I. TEORETICKÁ ČÁST

1 INFORMAČNÍ KIOSKY

Informační kiosky jsou jednoúčelové informační panely, které umožňují lidem vyhledat si určité informace pomocí dotykové obrazovky. Vyhledávání informací je umožněno připojením stanice do intranetu nebo internetu. Dá se říct, že informační kiosek je speciální využití počítače. Nasazení najdou všude tam, kde se pohybují lidé.

Vyrábí se různé typy informačních kiosků podle způsobu využití. Kiosky lze rozdělit dle umístění na ***vnitřní*** a ***venkovní***. Z toho vyplývá nutnost zvýšeného krytí IP u kiosků pro venkovní použití. Při venkovním použití je nutné zajistit vandalismu odolné provedení citlivých částí. Proto je vhodné použít displej s bezpečnostním sklem, speciální klávesnici, kovový kryt, apod. Totéž platí pro kiosky bez stálého dohledu umístěné v rozlehlých budovách jako jsou vlaková nádraží a letištní terminály.

Speciálním typem kiosků jsou jakékoliv automaty (výdej jízdenek, platba parkování, bankomat).

Základní možnosti využití kiosků

- Bankomaty
- Objednávkové terminály restaurací, jídelen
- Informační systémy letišť, hotelů, nádraží
- Přístup na internet
- Knihovny, úřady, prezentační místnosti či jiná frekventovaná místa
- Při řízení průmyslových technologií a strojů
- Reklama (kiosky umístěné na veřejných místech nebo ve výlohách, promítání reklamních informací)
- Herní zařízení
- Výukové trenažéry

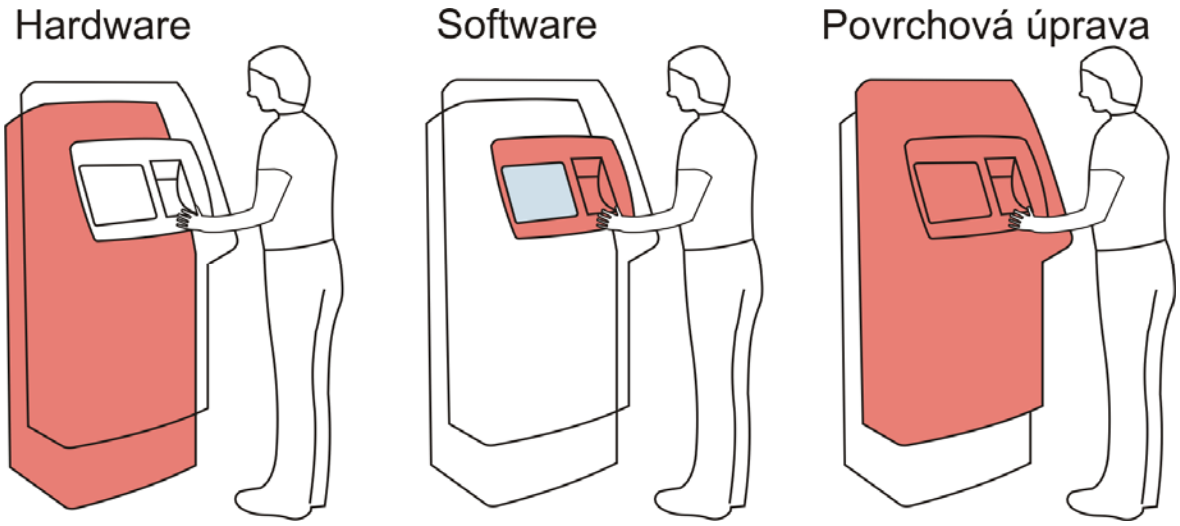
Toto všechno dnes dokáží kiosky zastat a díky kombinaci speciálních periferií mohou sloužit nejen jako informační zařízení, ale také jako zařízení poskytující služby a zboží.

Hlavní přínosy a přednosti kiosků

- Intuitivní ovládání bez znalostí výpočetní techniky
- Při vhodném umístění 24-hodinová dostupnost
- Interaktivní, snadno ovladatelný a přístupný zdroj informací
- Zaměření na různé cílové skupiny uživatelů
- "Svoboda" uživatele při získávání informací
- Nízké pořizovací a provozní náklady
- Nízká poruchovost
- Konfigurace na klíč (podle předpokládaného způsobu využití) [7]

1.1 Základní součásti informačních kiosků

Samotný informační kiosek lze rozdělit na tři základní části. První z nich je ***technické řešení*** (hardware), další částí je ***programové vybavení*** (software) a poslední je ***povrchová úprava*** (tvar, barva, …). Viz. následující obrázek.



Obrázek 1. Základní součásti informačních kiosků

Pro představu jsou uvedeny možnosti složení jednotlivých částí.

Hardwarové složení kiosků

- LCD displej s bezpečnostním sklem
- Klávesnice – s trackballem/bez trackballu, numerické klávesnice, touchpady
- Osobní počítač
- Tiskárna – termotiskárna, laserová tiskárna
- Čtečka karet
- Akceptor mincí a bankovek
- Ostatní – čtečka čárového kódu, sluchátko, reproduktory, webkamera

Programové vybavení

- Zabezpečený operační systém
- Bezpečný internetový prohlížeč – filtrování na základě URL nebo obsahu, znemožnění přístupu na stránky s nevhodným obsahem
- Firewall, ochrana proti spyware

Povrchová úprava

- Tvar – samostatně stojící, připevněný ke zdi
- Barva
- Antireflexní fólie na displeji

1.2 Dotykový displej

Dotykový displej (touchscreen, dotykový panel, dotyková obrazovka) je vstupní zařízení instalované na LCD nebo CRT displeje. Dotykové panely se čím dál častěji objevují a využívají všude tam, kde je potřeba zajistit snadné a rychlé ovládání a řízení přístroje či aplikace, nebo vytvořit intuitivní komunikační rozhraní PC i pro méně technicky zdatné uživatele.

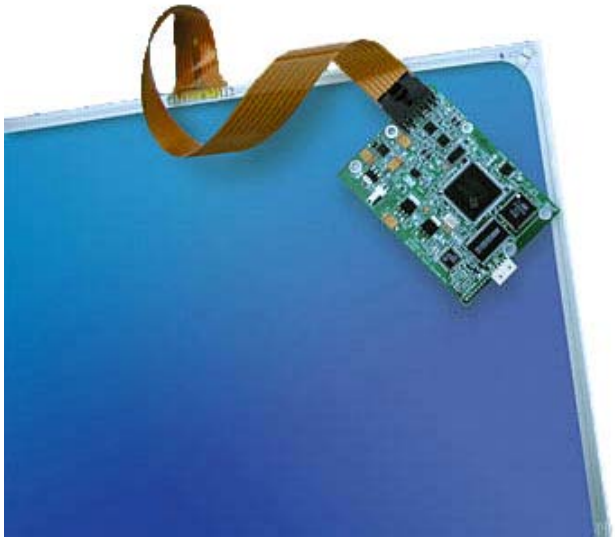
Dotykové panely mohou být snadno použity s většinou počítačových systémů a mohou plně nahradit primární vstupní zařízení, jako je klávesnice, myš či trackball, u nichž je nutná určitá zkušenost a znalost, které klávesy stisknout nebo jakým směrem a způsobem myší pohybovat.

V tom je velká výhoda dotykových displejů, stejně tak jako větší kompaktnost takového zařízení a jeho menší rozměry (odpadá místo pro klávesnici nebo jiné periferie). Naopak za zápory se dá považovat větší znečišťování displeje (zvláště při ovládání přímo prsty) a stínění zobrazení (například textu) v místě dotyku. Zpočátku může působit problémy lidem dlouhodobě zvyklým pracovat s klávesnicí a myší, což se projeví horší orientací v menu a programech.

Jak bylo zmíněno výše, dotykový panel je použitelný v široké škále aplikací. Vedle nich najdou panely uplatnění v zařízeních speciálně vyvinutých pro jedince, kteří mají problémy s ovládáním klasických periferií – klávesnice a myši. Lze je také uplatnit jako alternativa k běžnému používání myši například ve webovém prohlížeči. Aplikace vytvářené tzv. na míru pro tento druh displejů obvykle nabízí větší ikony, ovládací prvky (tlačítka, posuvníky,...) či odkazy než typické počítačové programy. Zvláště výhodné je pak použití dotykových displejů u přenosných zařízení, jako jsou kapesní počítače nebo mobilní telefony, protože tam je zvláštní klávesnice vždy problémem. [8]

1.2.1 Hlavní části dotykového displeje

Dotykový displej se skládá ze 3 hlavních částí - ***dotyková vrstva***, ***řadič*** (controller) a ***ovladač***. Řadič je malé zařízení, které řídí dotykovou vrstvu a spojuje ji s počítačem. Při dotyku vrstvy se vzniklý signál předá řadiči, který jej zpracuje a předá data procesoru. Jinými slovy řečeno převádí informaci z dotykové vrstvy na informaci, která je srozumitelná počítači. Ovladač je program, který komunikuje přes řadič s operačním systémem. Ovladač zajistí, že pohyb prstu je převeden na pohyb myší. [9]



Obrázek 2. Dotyková plocha s řadičem

Monitory s dotykovou vrstvou se vyrábí ve dvou základních variantách – dotyková vrstva jako součást obrazovky monitoru nebo samostatný rámeček s dotykovou vrstvou, který se dá připevnit ke klasickému monitoru.

Tento rámeček se skládá z vlastní dotykové plochy, řadiče a ovladače. Dotyková plocha je průhledná a je připojena do sériového nebo USB portu, případně do rozšiřující PCI karty.

1.2.2 Princip funkce

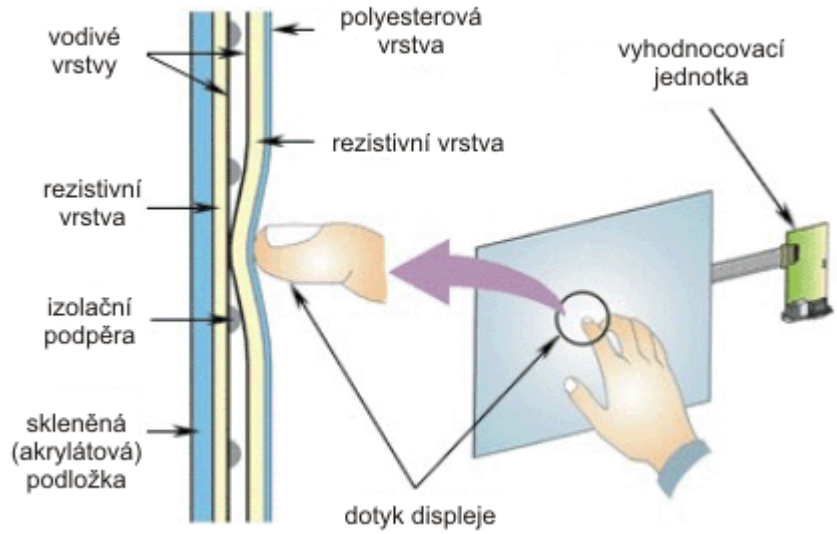
Běžné systémy dotykových obrazovek pracují na jednom z následujících principů:

- Rezistivní (odporový)
- Kapacitní (elektricky-citlivý)
- Akustický (povrchová akustická vlna)
- Infračervený (foto-citlivý) [10]

1.2.2.1 Rezistivní

Prvním z konstrukčních řešení dotykových displejů je tzv. rezistivní technologie. Takový displej se skládá ze dvou vrstev. Svrchní pružná polyesterová vrstva (membrána) je natažena ve vzdálenosti několika tisícin milimetru od spodního skleněného senzoru. Svrchní vrstva je na vnitřní straně pokryta velmi tenkou průhlednou vodivou vrstvou,

obvykle indium-tin-oxide (ITO). Také na vnější straně spodního skleněného senzoru je průzračná vodivá vrstva. Mezi vrstvami je pak velmi tenká vzduchová mezera s rastrem izolačních podpěr, které vodivé vrstvy izolují od sebe. Obě vrstvy jsou připojeny k řídicímu a vyhodnocovacímu modulu. Svrchní vrstva je pod napětím. Když se uživatel dotkne obrazovky prstem nebo jiným předmětem, pružná polyesterová vrstva se prohne a vodivě se spojí s vrstvou spodní. To vyvolá změnu elektrického napětí. Na základě velikosti této změny pak kontrolér vypočítá polohu bodu dotyku. Struktura rezistivního dotykového displeje je znázorněna na následujícím obrázku. [10]

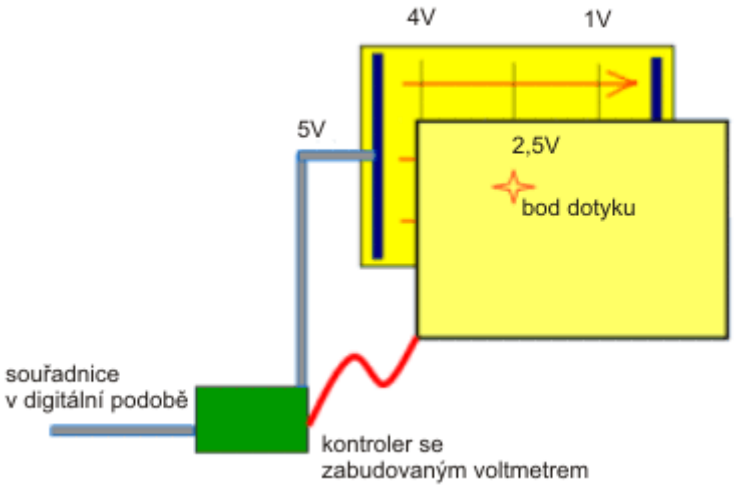


Obrázek 3. Struktura rezistivního dotykového displeje

Existuje několik druhů rezistivních dotykových displejů, které se rozlišují počtem "řídících" vodičů. Nejnižší možný počet vodičů je 4, další varianty jsou s 5, 6, 7, 8 vodiči, které nabízejí vyšší přesnost snímání a také delší životnost, která se odvíjí od počtu dotyků na displej. Například 6-ti vodičový dotykový displej má životnost 35 milionů dotyků v jednom místě. Životnost také závisí na typu použitého ukazovátka – prst je nejšetrnější (4-vodičový zvládne 5 milionů dotyků), při použití pera se životnost sníží na přibližně 1 milion dotyků.

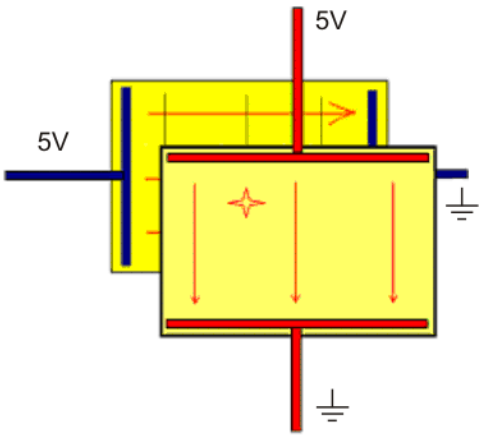
Na následujících řádcích je detailněji vysvětlen princip funkce 4-vodičového rezistivního dotykového displeje.

Čtyř-vodičová rezistivní technologie používá vrchní a spodní odporovou vrstvu k určení X a Y souřadnice místa dotyku. Jak lze vidět na následujícím obrázku, na jednu stranu spodní vrstvy je přivedeno napětí 5 V. Protilehlá strana je uzemněna a napětí v tomto místě je nulové. Napětí mezi jednou a druhou stranou vrstvy se mění lineárně. V klidovém stavu je napětí na vrchní vrstvě nulové.



Obrázek 4. Princip rezistivní technologie

Při dotyku se vodivé vrstvy spojí a velikost napětí na vrchní vrstvě odpovídá napětí dolní vrstvy v místě dotyku (v tomto případě 2,5 V). Řadič pomocí ADC (analog-to-digital converter) vyhodnotí velikost napětí a přepočte na X souřadnici. Poté se funkce obou vrstev prohodí a tím se získá Y souřadnice dotyku. Tzn., že na jednu stranu vrchní vrstvy je přivedeno napětí 5 V a na spodní vrstvě je napětí nulové. [11]

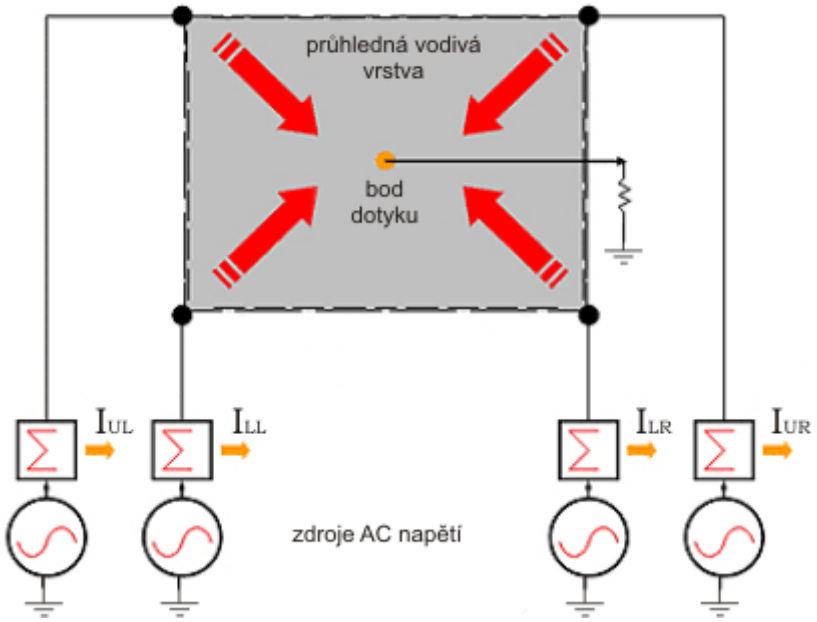


Obrázek 5. Zapojení vrstev pro výpočet X a Y souřadnice

Výhodou tohoto řešení je především to, že k dotyku lze použít prakticky cokoli. Může to být špička prstu a třeba i v rukavici, tužka nebo jakýkoli jiný předmět. Jde tu v podstatě jen o vyvinutý tlak na horní vodivou vrstvu. Nevýhoda těchto displejů spočívá ve zranitelnosti horní pružné vrstvy. Časem se na více používaných místech objeví průhyby, které poškodí tenkou vodivou vrstvu. Výsledkem je zhoršená přesnost určení místa dotyku. Dále hrozí poškození takového displeje ostrými předměty.

1.2.2.2 Kapacitní

Na povrchu kapacitního dotykového displeje je umístěna vodivá vrstva, která uchovává elektrický náboj. V rozích panelu jsou umístěny elektrody, které přivádí nízké napětí do vodivé vrstvy a vytváří homogenní elektrické pole. Při dotyku displeje prstem nebo jiným vodivým předmětem, začnou z elektrod proudit elektrické proudy, které jsou úměrné vzdálenosti místa dotyku od elektrod. Jinými slovy, část náboje přejde do ruky a dojde ke snížení náboje na kapacitní vrstvě. To způsobí změnu frekvence oscilačních obvodů připojených k elektrodám v závislosti na místě dotyku. Na základě změn frekvencí jsou vypočteny souřadnice dotyku. [12]



Obrázek 6. Princip kapacitní technologie

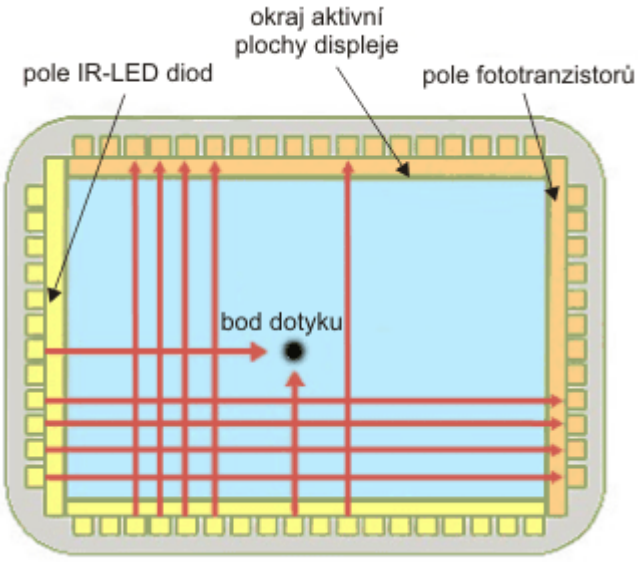
Výhodou této technologie je, že propouští téměř 90% světla vyzařovaného monitorem. Naproti tomu rezistivní dotyková vrstva propouští pouze 75 % světla. Mezi klady se také řadí vysoká mechanická odolnost (umožňuje 300 milionů dotyků v jedné oblasti) a velmi malá náchylnost na poruchy funkce vlivem ušpinění (mastnota, prach apod.). Naopak velikou nevýhodou a omezením je to, že dotyk displeje funguje jen v případě, že se obrazovky dotýká elektricky vodivý předmět.

Speciálním případem kapacitního displeje je pak tzv. ***projekční kapacitní displej***. Využívá principu kapacitního displeje, ale s tím rozdílem, že vyzařuje elektrické pole do blízkého okolí. Pokud je takový displej umístěn např. za nevodivou tenkou vrstvu skla nebo plexiskla, bude tento systém fungovat a bude přitom vysoce mechanicky odolný. [13]

1.2.2.3 Infračervený

Tato technologie je založena na přerušení toku světelného paprsku infračerveného záření. Kolem displeje je umístěn rám, který na jedné straně obsahuje řadu světelných zdrojů (IR-LED diody) a na protější straně se nachází světelná čidla (fototranzistory).

IR-LED diody vytvoří hustou síť neviditelných infračervených paprsků. Při dotyku displeje jakýmkoliv předmětem se světelný paprsek v tomto místě přeruší. To způsobí, že na světelné čidlo nedopadne žádný paprsek. Pomocí elektroniky se zjistí přesné X a Y souřadnice dotyku. [12]



Obrázek 7. Princip infračervené technologie dotykového displeje

Technologie je určena pro velmi náročné aplikace. Je to jediná technologie, která nemusí spoléhat na povrchový senzor pro registraci doteku. Pro aktivaci některého bodu displeje není nutné se dotýkat přímo podkladu a opotřebení dotekového skla je výrazně nižší. Další výhodou je 100% propustnost světla a také to, že diody a fototranzistory jsou umístěné za okraji displeje a jsou tak chráněny proti poškození. Životnost samotného snímacího mechanismu je neomezená. Nevýhodou je vysoká cena a menší rozlišovací schopnost.

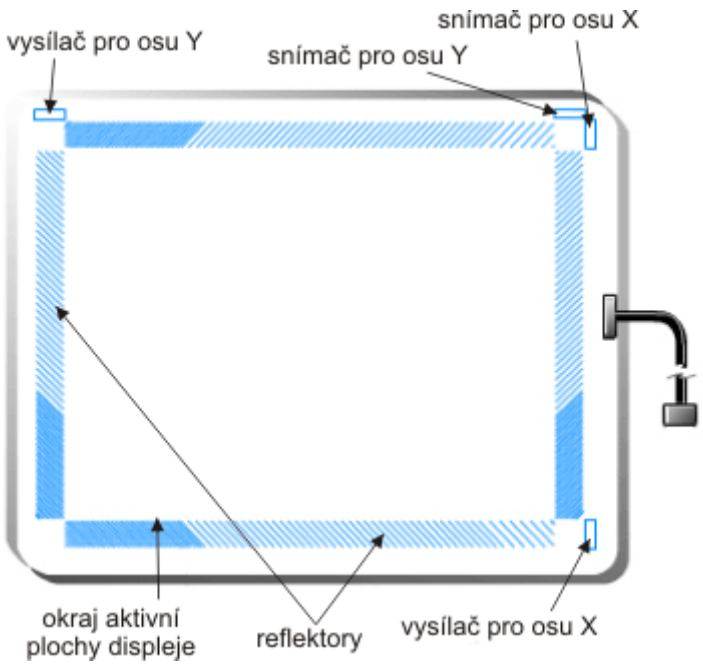
Takový systém lze zhotovit jako rám, který pak lze nasadit na jakýkoli monitor. Pokud je takto vybaven třeba nějaký starý CRT monitor, rázem se z něj stane moderní dotyková obrazovka. Příklad takového přídavného zařízení je na následujícím obrázku. [14]

Obrázek 8. Nasazovací modul IR dotykového displeje

1.2.2.4 Akustický

Nejvíce sofistikovanou metodou řešení dotykových displejů je technologie využití povrchové akustické vlny (ultrazvukové). Pro tyto displeje se používá označení SAW (Surface Acoustic Wave). Metoda je založena na vysílání akustických vln po skleněném panelu a následně jejich příjmu. Vlny se šíří napříč po ploše displeje. Vložením předmětu do vlnového pole se absorbuje část akustického vlnění. Dojde ke změně vlnění a jednotka pak podle vyslaných a přijatých signálů vyhodnotí polohu vložené překážky.

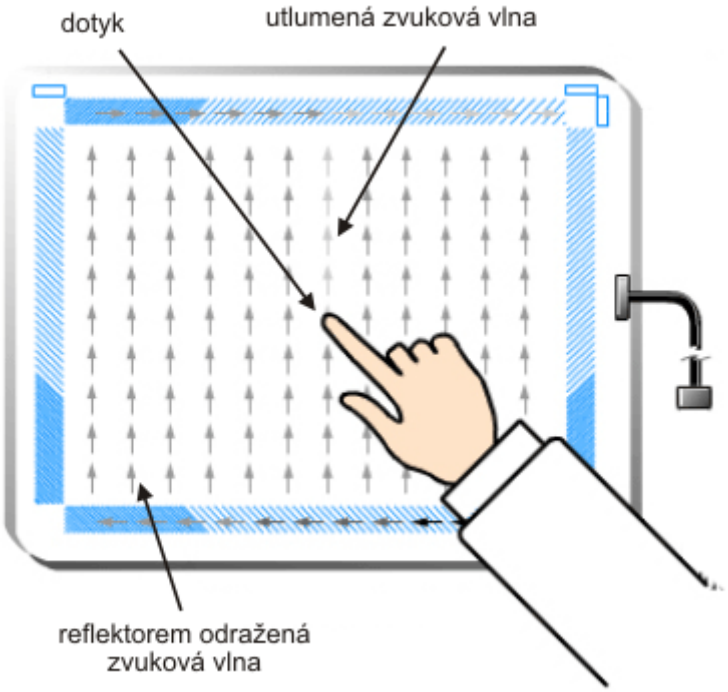
Na vrchní straně skleněné desky, obvykle speciálně vytvrzené a mechanicky odolné, se na jejím okraji nachází čtyři měniče a speciální reflexní prvky. Měniče, tvořené piezoelektrickými materiály, jsou umístěné v rozích desky a natočeny ve směru reflexních prvků. Dva měniče jsou obvykle vysílače, které převádějí elektrický signál na akustické vlny o frekvenci cca 5 MHz šířící se po povrchu skleněné desky. Zbylé dva měniče jsou pak přijímače, které převádějí akustickou povrchovou vlnu zpět na elektrický signál. Soustava reflexních prvků je tvořena sérií krátkých "proužků" orientovaných pod úhlem 45°, které odráží akustickou vlnu z vysílače napříč detekovanou aktivní plochou a na druhé straně pak zpět do prostoru přijímače. Pro odstínění všech vln, které se odrazí od reflexních prvků, jsou za nimi v rozích umístěny speciální pohlcující vrstvy. Tím se zamezí vzniku parazitních interferenčních jevů.



Obrázek 9. Popis základních prvků SAW dotykového panelu

Povrchové akustické vlny jsou na skle velmi snadno absorbovány měkkými materiály, které se ho dotýkají. Řídící elektronika vyrábí krátké vysokofrekvenční elektrické signály vždy pro jeden z vysílačů, který generuje akustické vlny. Na každém reflexním proužku se část této vlny odrazí a dále se pohybuje napříč detekovanou plochou. Na opačné straně jsou protilehlými reflexními proužky odráženy zpět do rohu, kde je umístěn přijímač, který vlny opět převádí na elektrický signál. Tento proces reprezentuje detekci v jedné ose (1D),

například v ose X, jak je ukázáno na obrázku 10. Použitím stejné struktury s dalším párem vysílač/přijímač a otočením o 90° se již vytvoří detekce v 2. ose, například Y a vznikne tak požadovaná 2D detekce plochy.



Obrázek 10. Princip SAW dotykového panelu

Cesta odražených vln je kratší pro trasy blíže k páru vysílač/přijímač a delší pro opačnou stranu displeje. Protože je rychlost šíření zvuku v rámci dotykové plochy konstantní a v porovnání s rychlostí zpracování dnešní elektroniky relativně pomalá, není problém pro řídící a zpracovávající obvody rozlišit jednotlivé dráhy napříč dotykovou plochou mezi sebou podle doby, kterou vlna potřebuje k jejímu uražení. Díky tomuto efektu je vyslaná energie rovnoměrně časově rozprostřena v přijímaném signálu (z krátkého impulsu z vysílače vznikne široký obdélník v přijímači) a lze tak rozlišit útlumy v jednotlivých dráhách a tedy i v místech displeje.

Vysílání i příjem je stále aktivní, přičemž se střídá detekce (skenování) v ose X a Y. Řízení si neustále vytváří a ukládá z přijímaného signálu "digitální mapu" dotykové plochy a tím umí případně eliminovat vliv trvalých tlumících prvků jako prach a nanesené nečistoty adaptivní změnou detekční hranice. Reaguje tak pouze na krátkodobé a výrazné útlumy

způsobené například dotykem prstu na skleněnou plochu. Pozice dotyku je rozpoznána z výrazného útlumu signálu v určité časové pozici v přijatém signálu v ose X a Y. Dotykem na sklo nějakým měkkým materiálem (prstem, koženou rukavicí, gumovým ukazovátkem apod.) dojde k absorpci dostatečného množství energie vlny, aby byla překročena v elektronice nastavená „spínací“ hranice. Z časové pozice naměřeného útlumu energie od počátku a konce příjmu se určí pozice dotyku na ploše. Digitální hodnoty souřadnic jsou pak již vyslány ke zpracování do elektroniky displeje.

Protože je panel celoskleněný a nejsou na něm přidány žádné vrstvy, které se dají mechanicky poškodit, je tato technologie mnohem odolnější než technologie odporová. Čistě skleněný povrch dotykového panelu umožňuje technologii 100% propustnost světla. Výhodou je také větší rozlišovací schopnost snímání dotyků. Nevýhodou je pak nutnost použít k dotyku jen měkké předměty, protože tvrdá ukazovátka jako je například tužka nefungují. Problematická je na této technologii vysoká citlivost na znečištění, protože i malá nečistota dokáže pohltit akustické vlnění a na displeji tak vznikají hluchá místa. Uváděná životnost je 50 milionů dotyků v jednom místě. [8]

1.2.3 Srovnání technologií

Těžko rozhodovat, která z technologií je ta nejlepší. Jsou tak různorodé, že by takové posuzování snad ani nemělo smysl. Každá ze zmíněných technologií je vhodná pro jiné využití. Některé jsou vhodnější do drsných podmínek průmyslové výroby a používají se proto na ovládání a programování výrobních robotů, jiné jsou oproti tomu méně robustní, ale o to více přesné v určování polohy.

Závěrem malé srovnání jednotlivých technologií (vychází z tabulky uvedené [12]).

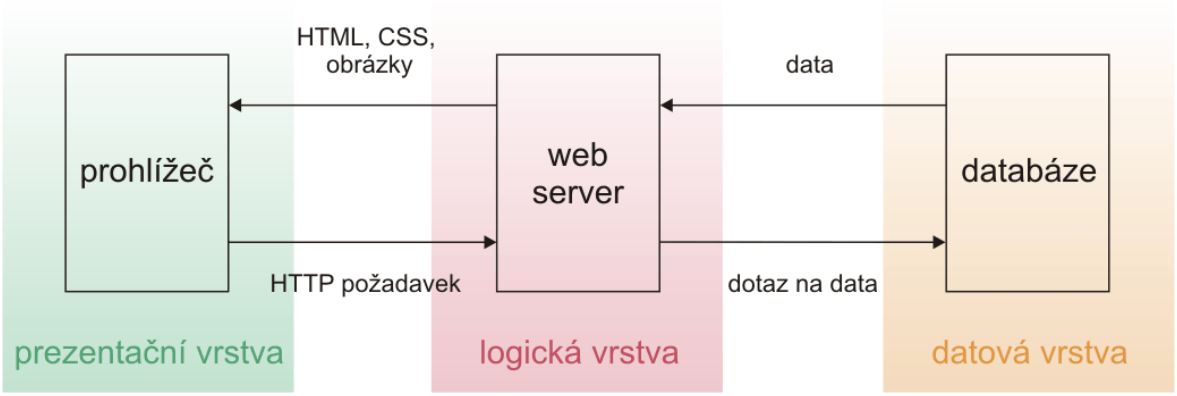
Tabulka 1. Srovnání dotykových technologií

Technologie / vlastnost	Rezistivní	Kapacitní	Infračervená	Akustická
Rozlišení	vysoké	vysoké	průměrné	vysoké
Propustnost	průměrná (70-80%)	dobrá (90%)	vysoká (100%)	vysoká (100%)
Ukazovátko	prst nebo pero	pouze prst	prst nebo pero	prst nebo pero s měkkým hrotem
Odolnost	poškození ostrým předmětem	vysoká	vysoká	citlivé na prach a vlhkost

2 RICH INTERNET APPLICATION (RIA)

Rich Internet Application (doslovně přeloženo Bohatá internetová aplikace) je nový směr, kterým se ubírá další generace internetových aplikací. Tyto aplikace vynikají vysokou mírou interaktivity, inteligence, efektivity, snadnosti použití a flexibility. Technologií k vývoji RIA existuje hned několik. Jsou výsledkem spojení toho nejlepšího z prostředí desktopových i internetových aplikací.

Hlavním rozdílem mezi tradičními internetovými aplikacemi a RIA je fakt, že se data nemusejí nahrávat při každém dotazu – kliknutí nebo jiné uživatelovy interakce. Nahrají se jen jednou a pak už jsou k dispozici ihned, po celou dobu spuštění aplikace. Nebo se nahrávají průběžně na její pozadí, aniž by uživatele obtěžovala. [15]



Obrázek 11. Tradiční model webové aplikace

Současné aplikace musí řešit dva primární problémy. Za prvé, aplikace se musí srovnat s HTTP protokolem a jeho modelem žádost/odpověď. Sebemenší změna stavu (autokompletace dat, validace, aktualizace části dat) u klienta musí vyvolat požadavek na server, který jej musí obsloužit a zpět vrátit všechna předešlá data. Druhým problémem jsou poměrně omezené možnosti prezentačních technologií (HTML, CSS) pro kompaktnější grafická rozhraní. [16]

Pomocí RIA se technologie vyvinula do stavu, kdy překračuje omezení jazyka HTML a k internetovým aplikacím přidává i bohaté grafické rozhraní s plnohodnotným uživatelským zážitkem. Vznikají tak aplikace "bohatší" na uživatelský zážitek, tedy "Rich internet application".

Se současným stavem internetu jsou tyto aplikace dostupné prakticky všude a zároveň není uživatel frustrován čekáním na server při každém dotazu na novou stránku po každém kliknutí myši – ohromná výhoda oproti tradičním internetovým aplikacím.

Obrázek 12. Technologie RIA

Technologií k vývoji RIA existuje několik. Mezi nejpopulárnějšími jsou technologie založené na HTML, jako je AJAX (Asynchronous Javascript And XML). Nebo existují možnosti založené na plug-inech, jako jsou Adobe Flash, Adobe Flex a Laszo, které běží ve Flash Playeru. Další možností tvorby RIA aplikací je Windows Presentation Foundation od firmy Microsoft. [15]

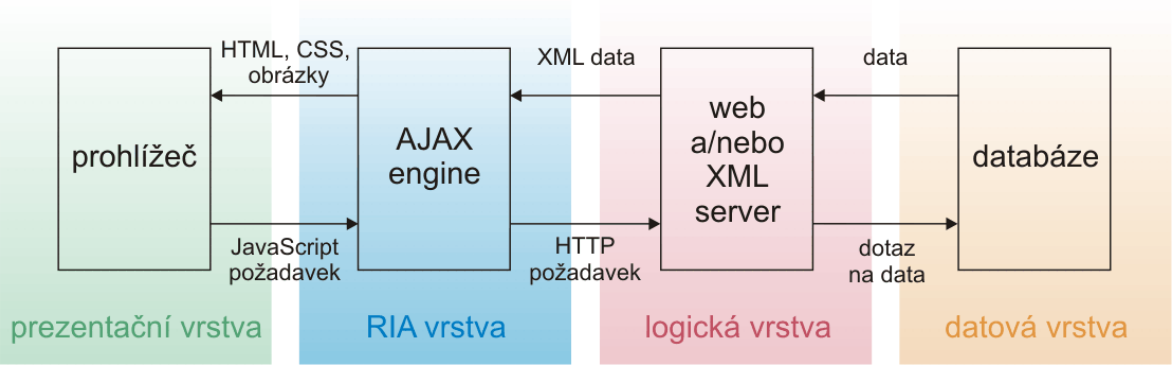
2.1 AJAX

AJAX (Asynchronous JavaScript and XML) je obecné označení pro technologie vývoje interaktivních webových aplikací. AJAX stojí na technologiích, které jsou webovým vývojářům dlouho známé: HTML, DHTML a JavaScriptu. Základní myšlenkou je použití JavaScriptu k aktualizaci stránky bez potřeby jejího načítání. To umožňuje javascriptový objekt **XMLHttpRequest**, který zavedl Microsoft v Internet Explorer 5. Poté byl tento objekt implementován do většiny moderních internetových prohlížečů. Nejčastěji se AJAX používá za účelem vylepšení uživatelské interakce, zefektivnění komunikace a zrychlení doby odezvy. Dnes se již jedná o používanou a široce rozšířenou techniku tvorby vysoce výkonných a efektivních webových aplikací. [17]

Není to programovací model sám o sobě, ale může využívat následující technologie. Technologie, které jsou podtrhlé, jsou nezbytně nutné:

- *HTML/XHTML* – pro prezentaci obsahu na internetu
- *CSS* – poskytuje stylistické formátování do XHTML
- *DOM* – dynamické aktualizace nahrávaných stránek
- *XML* – formát výměny dat
- *XSLT* – jazyk transformace XML dokumentů do (typicky) XHTML dokumentů (obvykle za použití CSS)
- *XMLHttpRequest* – pro asynchronní výměnu dat s webovým serverem (typicky je užíván formát XML, ale je možné použít libovolný jiný formát včetně HTML, prostého textu)
- *JavaScript* – skriptovací jazyk pro programování s Ajax engine [17]

V klasickém webovém modelu každá změna stavu u klienta vyžaduje obnovení celého uživatelského rozhraní, tzv. **http request-response přístup**. Nejdříve je tedy žádost o změnu stavu, odeslání požadavku na server, vyřízení požadavku a vše končí zasláním kompletního uživatelského rozhraní s daty, přičemž jednotlivé kroky jsou vzájemně synchronizovány. Naopak AJAX, díky XMLHttpRequest, může vyvolat libovolný počet nezávislých požadavků, jejichž výsledky mohou ovlivnit pouze patřičné části uživatelského rozhraní, bez nutnosti opětovného načítání celého HTML dokumentu. Změní se tedy jenom požadovaná část stránky a odstraní se tím také nepříjemné blikání při běžném přechodu mezi stránkami a v neposlední řadě také objem dat přenášený mezi serverem a prohlížečem. Internetové stránky tak už nemají jen funkci po sobě jdoucích stránek, ale díky větší plynulosti práce se více blíží běžným desktopovým aplikacím. [16]



Obrázek 13. Koncept AJAX aplikace

Další výhodou je fakt, že společně s používáním známých technologií při vývoji, AJAX ke svému chodu nepotřebuje žádné další plug-iny. Funguje s JavaScriptem a DHTML, které už prohlížeč obsahuje. Nicméně spoléhat se na JavaScript s sebou přináší nová rizika: aplikace při vypnutém JavaScriptu nefunguje.

Problém s Ajaxem je v podpoře různých verzí DHTML i JavaScriptu u rozličných prohlížečů a na různých platformách. V případech, kdy se dá cílová skupina kontrolovat (např. intranety), může být vytvořena aplikace k podpoře jednoho prohlížeče na určité platformě (mnoho společností má dnes sjednocené operační systémy a prohlížeče). Pokud je ale aplikace přístupná široké veřejnosti (extranetové nebo internetové aplikace), Ajaxové aplikace je potřeba testovat a vyvíjet na všech operačních systémech i všech prohlížečích. [15] [16]

2.2 Adobe Flash

Další možností tvorby aplikací v prostředí RIA je Flash platforma od společnosti Adobe (dříve Macromedia), v současnosti klíčový konkurent AJAXu.

Flash byl původně technologií určenou k vytváření složitých animací. Ve svých počátcích býval Flash používán pouze při tvorbě reklamních bannerů, pro které je vhodný i díky malé velikosti souborů (využívá vektorovou grafiku). Dnes má však Flash mnohem rozsáhlejší možnosti využití – hodí se ke tvorbě prezentací, vytváření on-line her či interaktivních aplikací. [19] To umožňuje programovací jazyk ActionScript (v současné době ve verzi 3.0). Samotný program se skládá z pracovního prostředí (zde uživatel kreslí

objekty a umísťuje je do časové osy) a vývojového prostředí, kde se nachází ActionScript editor.

Obrázek 14. Logo Adobe Flash CS3

Významnou výhodou technologie Flash je možnost uložit hotovou aplikaci i jako spustitelný soubor a distribuovat ji tak i offline např. na CD-ROM. Stále větší význam dostává také Flash Lite, mladší a mírně odlehčená verze určená pro mobilní zařízení. [19]

Tím však přednosti technologie Flash zdaleka nekončí. Podobně jako technologie AJAX bývá Flash používán k vytváření RIA aplikací, těží zde ze schopnosti přehrávat audio a video, pracovat s databázemi a dalšími pokročilými nástroji.

Pro běh interaktivních aplikací vytvořených ve Flashi je nutné do webového prohlížeče doinstalovat plug-in – Flash Player. Jak se zdá, pro uživatele to není žádný problém, protože podle statistik má v současnosti Flash Player u české internetové populace "pokrytí" přes 90%. [20] V opačném případě prohlížeč nabídne stažení aktuální verze Flash Playeru, viz. obrázek.

Obrázek 15. Nabídka stažení Flash Playeru

Flash Player se během let vyvinul z prostředí původně vytvořeného pro přehrávání animací. S každou novou verzí se ale přidávají nové možnosti, přičemž stále zůstává zachována malá velikost Flash Playeru. Macromedia (nyní Adobe) se od roku 2002 začala na Flash zaměřovat více než jako na animační nástroj. S uvedením verze Flash 6 začala

poskytovat více možností pro tvorbu aplikací. Ukázalo se, že kombinací všudypřítomného přehrávače a síly programovacího jazyka ActionScript, mohou vývojáři vytvořit plnohodnotnou aplikaci v prohlížeči bez omezení HTML. [15]

Obrovskou výhodou Flash Playeru je skutečnost, že je multiplatformní, tzn. obsah i samotná aplikace vyvinutá pro jeho určitou verzi bude fungovat ve všech prohlížečích a platformách, které tu konkrétní verzi Flash Playeru podporují.

Flash Player se stal také klientem RIA pro komerční framework Adobe Flex a open source framework OpenLaszlo s komerční podporou firmy Laszlo Systems.

Obecně je možné konstatovat, že RIA postavené na Adobe Flash Player nabízejí mnohem větší možnosti pro řešení komplexností grafického rozhraní a uživatelského komfortu oproti řešení postavenému na AJAXu. Dokonce, i když je toto téma vždy diskutabilní, lze dosáhnout většího oddělení aplikační a prezentační logiky, a tím dosáhnout také větší efektivity vývoje. [15] [16]

2.3 Adobe Flex

Detailnější popis této RIA technologie je uveden v praktické části této práce, v kapitole 7.

2.4 Adobe AIR

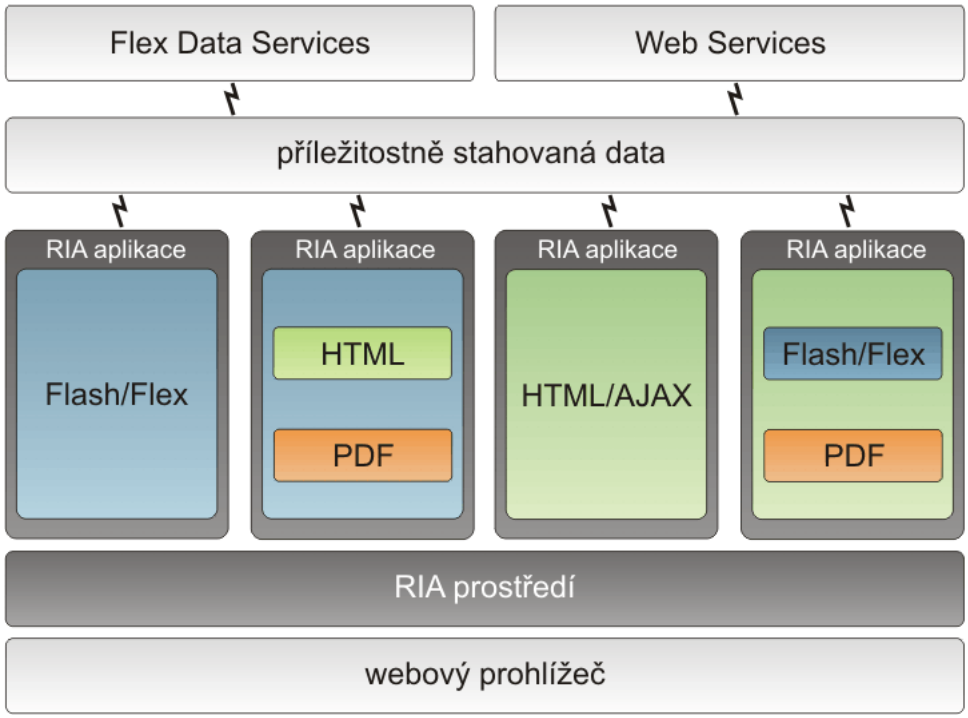
Nové multiplatformní prostředí Adobe AIR (Adobe Integrated Runtime) bylo po několika beta verzích uvolněno v ostré verzi koncem února letošního roku. Původně byla tato technologie označována kódovým názvem Apollo. Jedná se o prostředí pro spouštění RIA aplikací, zaručující konzistentní vzhled a kompatibilitu s většinou operačních systémů. Na rozdíl však od výše zmíněných technologií, AIR budou vyvíjené jako běžné aplikace. Neztratí se však ale žádné výhody RIA aplikací. K jejich vývoji se budou používat zavedené technologie (Flash, Flex, HTML, JavaScript, AJAX a PDF) a k internetu se budou připojovat přímo, bez potřeby internetového prohlížeče. [15]

Vývojářům publikujícím aplikace pro zmíněné prostředí se tak naskýtá výborná možnost, jak přenést svoje projekty z webových prohlížečů na úroveň běžných desktopových aplikací, které mohou bez větších potíží pracovat se soubory, databázemi, spolupracovat s jinými aplikacemi na počítači uživatele, nebo komunikovat po síti.

Obrázek 16. Logo Adobe AIR

Hlavní části Adobe AIR mají open source zdrojový kód, včetně modulu **Webkit HTML**, virtuálního počítače pro ActionScript (projekt **Tamarin**) a funkčnosti lokální **databáze SQLite**. Navíc společnost Adobe vydala Adobe Flex SDK (software development kit) jako open source.

Následující obrázek je převzat ze článku [21].



Obrázek 17. Aplikace Adobe AIR mohou využívat Flash, Flex, HTML/AJAX i PDF

O běh těchto aplikací se stará AIR přehrávač, který je obdobou Flash Playeru. Stejný přehrávač je k dispozici pro Windows, MAC OS a v nejbližší době bude i pro Linux. Z uvedeného plyne, že před instalací AIR aplikace si uživatel musí nainstalovat samotné prostředí, výhodou ovšem je to, že pomocí Flash Playeru je možnost detekovat, zda uživatelův systém AIR aplikace podporuje, nebo potřebuje novou instalaci.

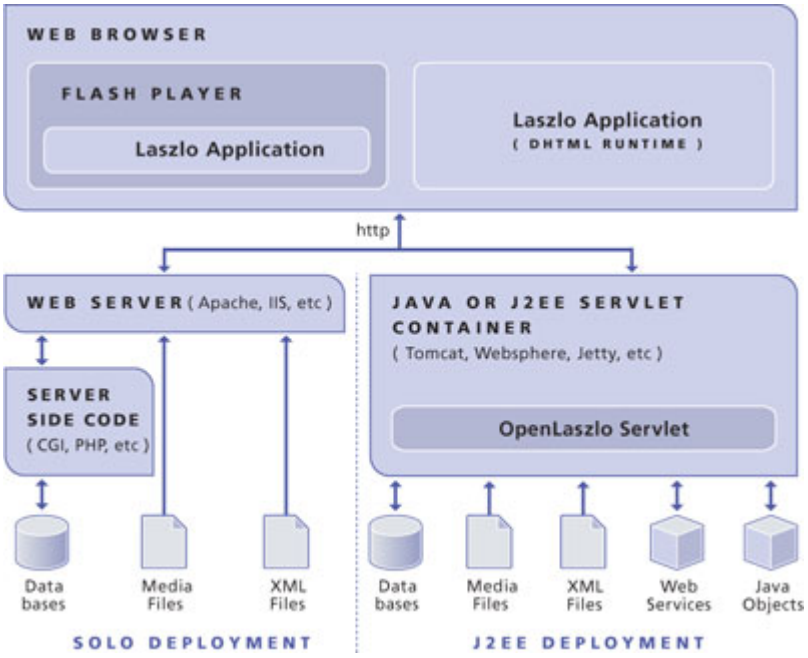
Mezi zajímavé funkce AIR patří:

- Podpora HD videa a kodeků H.264
- Hardwarová akcelerace při přehrávání
- Webový prohlížeč založený na jádru Webkit, který už umí renderovat i flashové animace v přehrávaných webových stránkách
- Možnost práce s databázemi (SQLite)
- Automatická instalace a podpora automatického stahování nových verzí
- Poslední verze Flash Playeru umožňuje detekovat AIR prostředí
- Podpora drag'n drop interakce se systémem
- Možnost registrace typů soborů, plná práce se soubory/adresáři
- Pro Flex Builder 3 je k dispozici celá řada komponent (tree menu adresářové struktury, webový prohlížeč, komponenty práce se soubory) [22]

2.5 OpenLaszlo

OpenLaszlo je open-source platforma, která umožňuje stejně jako AJAX vyvíjet aplikace postavené na JavaScriptu (přesněji ECMAScriptu) a XML, ale běžící ve Flash Playeru. Na rozdíl od Ajaxových aplikací Laszlo používá kompiler, který z JavaScriptu a XML vytvoří Flash SWF soubor. To vývojářům dovoluje osvobodit se od omezení různých verzí JavaScriptu napříč rozličnými platformami. Novinkou od verze 4.0 je možnost kompilace aplikace do DHTML. K dispozici je ke stažení IDE založené na Eclipse. [23]

Architektura frameworků Adobe Flex a OpenLaszlo je víceméně stejná, za pozornost ovšem stojí Laszlo Presentation server (obdoba Flex server). Flex i OpenLaszlo řeší, narozdíl od AJAXu, serverovou část – z hlediska třívrstvé architektury sedí na vrcholku prezentační vrstvy. Serverová část tak může držet stavy jednotlivých klientských komponent, zajišťovat komunikaci v datově efektivním formátu a umožnit napojení aplikační logiky na jednotlivé události uživatelského rozhraní. Od uvedení OpenLaszlo servletu pracují Laszlo aplikace také s webovými službami SOAP, XML-RPC a JavaRPC. [16]



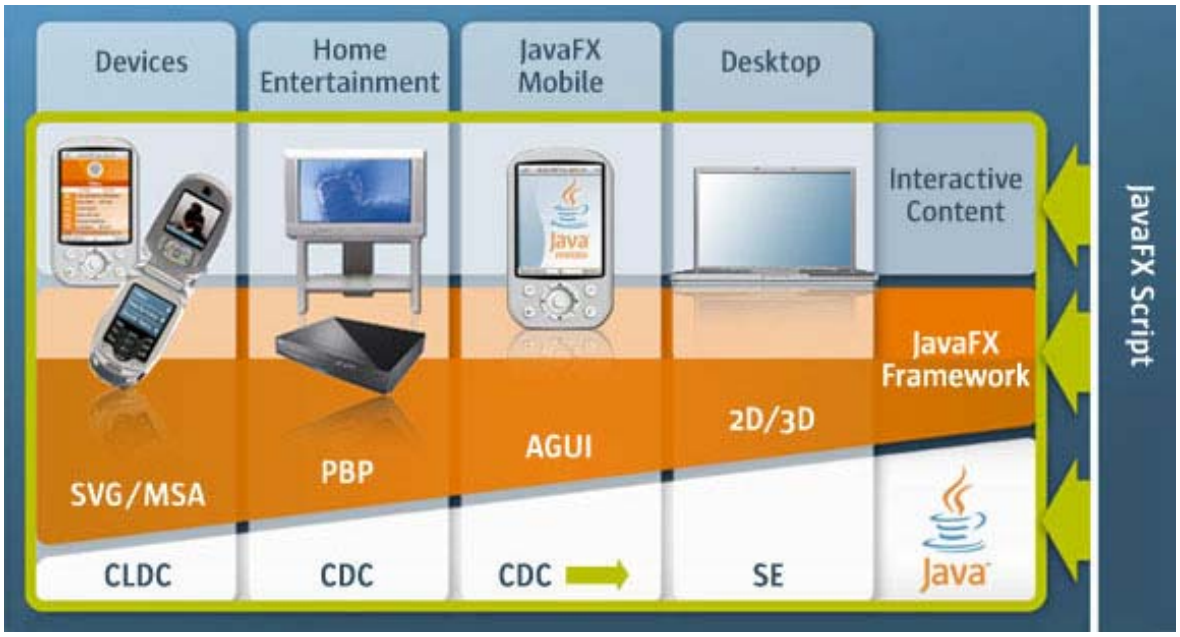
Obrázek 18. Architektura OpenLaszlo

Zdroj: http://www.openlaszlo.org/files/architecture.jpg

2.6 JavaFX

Novinkou z dílny Sun MicroSystems je JavaFX. Je poskytována pod GPL licencí. Funguje všude tam, kde běží Java Runtime. Základem je jazyk ***JavaFX Script*** - původní jméno F3. Je to staticky typovaný deklarativní scriptovací jazyk, zaměřený především na oblast Rich Internet aplikací a pokrývající mobilní zařízení (tzv. JavaFX Mobile), desktopové počítače a aplikace pro TV set-top boxy. [25]

Výhodou JavaFX aplikací je možnost jejich spouštění v offline módu. Tyto aplikace splňují heslo ***write once and run anywhere*** (napiš jednou a spusť kdekoliv). Stačí doinstalovat malou knihovnu a stejný skript poběží na všech přístrojích (PC, mobilních telefonech, digitálních TV), kde je nainstalována Java SE nebo Java ME. Odpadá tak problém s kompatibilitou javascriptového kódu pro různé prohlížeče, jak je tomu u AJAXu. [16]



Obrázek 19. Možnosti využití JavaFX Script

Základní informace a ukázka JavaFX aplikací z konference JavaPolis 2007 je ke shlédnutí na stránce: http://www.parleys.com/display/PARLEYS/Introduction+to+JavaFX

2.7 Microsoft Silverlight

Systém Silverlight od Microsoftu, donedávna označovaný WPF/e (Windows Presentation Foundation Everywhere), je postavený na frameworku .NET a evolučně vychází z technologie Windows Presentation Foundation (WPF). Silverlight je přímou konkurencí Flexu. Základem je značkovací jazyk XAML (založený na XML), který umožňuje popisovat textový obsah, grafický bitmapový a vektorový obsah a multimediální obsah (např. streamování videa, mp3, wma). Jazyk XAML (eXtensible Application Markup

Language) slouží pro definování prezentačního rozhraní aplikace (designu) a je možné ho využít také k definici systémových zdrojů nebo celé aplikace. [26] [27]

SilverLight aplikace se skládá ze tří částí:

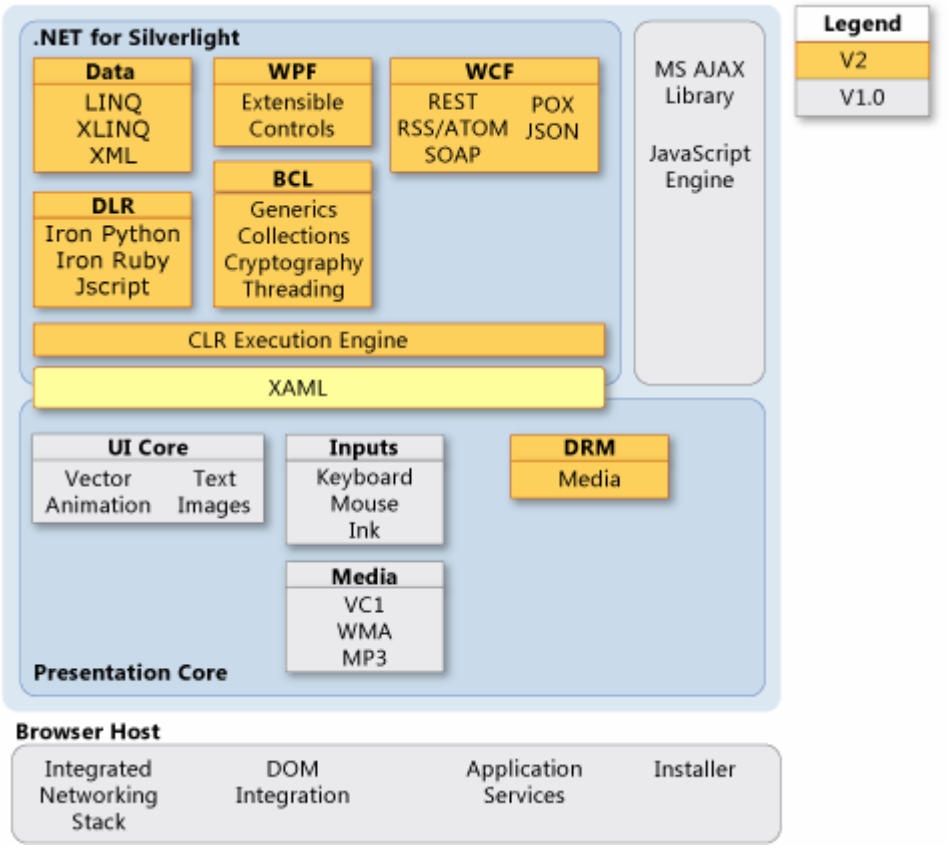
- XHTML stránka – hostitel Silverlight aplikace
- XAML soubor – definice uživatelského rozhraní
- Soubory s aplikační logikou

Aplikační logika a interaktivita – to znamená obsluha událostí (zobrazení stránky, stisknutí tlačítka nebo jiného ovládacího prvku, napsání znaku, pohyb kurzoru myši nad nějakým objektem…) je ve verzi Silverlight 1.0 zajištěná programovým kódem v jazyku JavaScript. Druhá verze přináší podporu kompilovaných .NET jazyků (např. C# a VB.NET). Jejich běh bude zajišťovat CLR integrovaný v plug-inu. Ze skriptovacích jazyků jsou podporovány vedle Javascriptu také Python a Ruby. Jejich běh zajišťuje DLR. Díky projektu Phalanger, což je implementace PHP pro .NET, je dalším podporovaným jazykem také PHP. [28]

Aplikační logiku je tedy možné oddělit od designu (v jednom souboru je XAML kód a v druhém logika napsaná v některém z uvedených jazyků).

Je zde zabudovaná podpora webových služeb. Silverlight podporuje LINQ a LINQ-to-XML a k veškerým datům je možno přistupovat přes RSS, REST, JSON či POX. Dalším trumfem je podpora videa v HD kvalitě díky kodeku VC-1, který využívá HD DVD i Blu-ray. [29]

Na následujícím obrázku je zobrazena architektura Silverlight ve verzi 2.0. Novinky, které tato verze přináší, jsou barevně odlišeny. Šedá barva reprezentuje možnosti verze 1.0, žlutou barvou jsou zvýrazněny technologie přístupné od verze 2.0. Obrázek s detailním popisem, se nachází na stránce [30].



Obrázek 20. Architektura Microsoft Silverlight 2.0

Zdroj: http://msdn2.microsoft.com/en-us/library/bb404713.aspx

Detailnější ukázka možností Silverlight a jeho architektury je plakát, umístěný na stránkách firmy Microsoft:

http://download.microsoft.com/download/f/2/e/f2ecc2ad-c498-4538-8a2c-15eb157c00a7/SL_Map_FinalNET.png

Silverlight se instaluje do nejpoužívanějších webových prohlížečů (v současnosti Internet Explorer, Firefox, Safari) ve formě rozšíření (plug-in) a pro spouštění Silverlight aplikací na klientském počítači není vyžadována instalace .NET Framework. Podporovány jsou operační systémy Windows Vista, Windows XP SP2 a Apple Mac OS X. Připravuje se oficiální podpora operačního systému Windows 2000 a prohlížeče Opera. Výhledově se počítá s podporou operačního systému Linux (plug-in s názvem Moonlight). [26]

Obrázek 21. Logo Microsoft Silverlight

Využití této technologie je velmi široké. Lze ji použít k tvorbě animovaných her včetně 3D grafiky, inteligentních reklam, elektronických knih, přehrávačů multimedií, ale taky k tvorbě rozsáhlých aplikací pro střih videa a grafických editorů. Samozřejmostí jsou menu začleněná do klasických webů.

3 CSS - KASKÁDOVÉ STYLY

CSS (Cascading Style Sheets) neboli kaskádové styly, vznikly jako souhrn metod pro úpravu vzhledu stránek. První návrh normy byl zveřejněn v roce 1994, v roce 1996 byla vydána specifikace CSS 1, v roce 1998 CSS 2 a nyní se pracuje na verzi CSS 3. CSS je vyvíjen jako standard konsorciem World Wide Web Consortium (W3C), více [31].

CSS se využívá k formátování obsahu HTML, XHTML a XML dokumentů a umožňuje oddělit styl stránky od jejího obsahu. Ve srovnání s formátováním pomocí atributů v HTML formátovací schopnosti rozšiřuje. Styly umožňují přesně určit, jak který element bude vypadat. Narozdíl od atributů je stylem možné definovat jednotný vzhled elementu v celém dokumentu (např. že všechny nadpisy úrovně 1 budou červené) a to jediným zápisem pro příslušný element (nikoli v každém tagu příslušného elementu). Stejně tak je možné pomocí stylu určit odlišné formátování pro třeba jen jediný výskyt určitého elementu. Tím se lze jednak zbavit velkého množství kódu a navíc se tento kód stane mnohem přehlednější. Změnit například barvu písma všech odstavců, je otázkou několika málo vteřin, měnit každý atribut u každého elementu v HTML by byla katastrofa. Jeden styl je možné snadno použít pro libovolné množství stránek. [32]

Styl se může nadeklarovat třemi způsoby [33]:

- **Přímý zápis** u formátovaného elementu pomocí atributu style="...".

```
<p style="color: red">Červený odstavec.</p>
```

- **Stylopisem** (Stylesheet) v hlavičce dokumentu. Stylopis je seznam stylů, které jsou uzavřené mezi tagy `<style></style>`.

```
<style>
   p {color: red}
</style>
```

- **Externím CSS souborem**, na který se stránka odkazuje tagem `<link>`. V souboru je umístěný stylopis. Hlavní výhoda je v tom, že na jeden takový soubor se dá nalinkovat mnoho stránek, takže pak všechny vypadají podobně.

```
<link rel="stylesheet" type="text/css" href="styly.css">
```

3.1 Selektor a deklarace

Styl se skládá z pravidel pro jednotlivé elementy, které mají být formátovány. Každé takové pravidlo má dvě části, ***selektor*** (název elementu, pro který má toto pravidlo platit) a ***deklaraci*** (co pro něj má platit). V deklaraci se určuje vlastnost a její hodnota, deklarace je uzavřena do složených závorek. Celé se to zapisuje takto:

```
selektor {vlastnost: hodnota_vlastnosti}
```

Například selektorem, tedy elementem, který je potřeba formátovat, je `h1` (nadpis 1. úrovně). Deklarace se zvolí `{color: blue}`. Ta určuje, že vlastnost `color` bude mít hodnotu `blue`. Celé dohromady to tedy znamená, že všechny nadpisy 1. úrovně v dokumentu budou mít modrou barvu. Zápis vypadá následovně:

```
h1 {color: blue}
```

Je možné určit elementu více než jednu vlastnost, jednotlivé vlastnosti se od sebe oddělí středníkem. Takto se může definovat libovolné množství vlastností. Samozřejmě je možný i zápis každé vlastnosti zvlášť, ale to je zbytečné.

```
selektor {vlastnost1: hodnota_vlastnosti1;
    vlastnost2: hodnota_vlastnosti2;}
```

Společná vlastnost pro dva elementy se určí tak, že se jednotlivé selektory oddělí od sebe čárkou.

```
selektor1, selektor2 {vlastnost: hodnota_vlastnosti;}
```

U většiny vlastností se využívá principu ***dědičnosti***. To znamená, že element, který nemá vlastnost definovanou, ji dědí po nadřazeném elementu. Týká se to především vlastností písma – barvy, velikosti, stylu atd. Vlastnost, kterou budou mít všechny elementy společnou, je vhodné definovat v elementu body (a později je případně možné vytvářet výjimky).

Pokud se ve stylu definuje pro stejný element stejná vlastnost dvakrát, vyšší váhu má ta deklarace, která byla definovaná později (myšleno na pozdějším řádku), a ta se také provede. K přiřazení větší důležitosti některé deklaraci se používá `!important`. [32]

```
h1 {color: blue !important}
```

Na webové stránce [32] v kapitole *IX. CSS vlastnosti – přehled* je uveden kompletní přehled vlastností CSS1 a některé vlastnosti a hodnoty CSS2.1.

3.2 Třídy a identifikátory

Třídy a identifikátory v CSS slouží k tomu, aby se mohly různé elementy formátovat různě. Například odkazy na stránce. Každý webdesigner chce asi mít na stránce různé druhy odkazů, ne jen jeden. Jinak se obvykle dělají odkazy v menu, jinak odkazy v textu.

Třídy se vytvoří snadno tak, že se k elementu v HTML přidá atribut `class`. Jeho hodnotou bude nějaký řetězec písmen, stejný se pak bude používat v CSS stylu jako selektor.

```
<p class="poznamka">Nějaký text</p>
```

Tímto je řečeno, že tento odstavec bude formátován podle pravidel třídy `poznamka`, na formátování ostatních odstavců se tato pravidla neprojeví. Teď se ještě musí ta pravidla určit v CSS stylu.

```
.poznamka {font-size: x-small; color: black}
```

Nyní budou všechny odstavce stejné, jen odstavec s třídou `poznamka` bude vypadat jinak (malým černým písmem).

Dále je možné přiřadit styl určitému elementu. Níže uvedený styl se aplikuje na elementy `li` s třídou `poznamka`. Jako první je uveden zápis v CSS, následuje HTML kód.

```
li.poznamka {color: blue}
```

```
<li class="poznamka">Položka seznamu</li>
```

Identifikátor se od třídy liší tím, že se vždy jedná o jednoznačný identifikátor. To znamená, že se může na každé stránce použít jen jednou, v rámci celého webu se může stejný identifikátor použít libovolněkrát. Oproti tomu třídu je možné použít libovolněkrát na každé stránce webu.

Identifikátory se tedy používají právě tam, kde je jisté, že se daný element objeví ve stránce jen jednou. Ideálně se tedy hodí pro věci jako je box celé stránky, menu, záhlaví

nebo zápatí. Identifikátory se označují dvojkřížkem (#). Jinak je jejich zápis stejný jako zápis třídy. Zápis v HTML kódu

```
<div id="menu"> … </div>
```

a definice v CSS

```
#menu {width:14em; background-color:black}
```
[32]

3.3 CSS validátor

CSS validátor slouží k validaci CSS stylu. Tzn. najde chyby, které CSS styl obsahuje. K validaci lze použít **validátor W3C**, nachází se na adrese *http://jigsaw.w3.org/css-validator*.

Co je možné validovat:

- ***CSS styl z Internetu*** zápisem URL adresy stylu do políčka *Validate by URI* a stisknutím tlačítka *Check*.
- ***CSS styl uložený na disku počítače*** přímým zápisem cesty k souboru na lokálním disku do políčka *Validate by File Upload* nebo výběrem s procházením disku pomocí tlačítka *Browse* a stisknutím tlačítka *Check*

Pokud je stylopis napsán správně (podle CSS specifikace), zobrazí se hláška *No error or warning found a Congratulations!*. Následuje kód pro přidání CSS validní ikonky a výpis validovaného CSS stylu. [32] V opačném případě se zobrazí chybová hláška podle druhu nalezené chyby.

Stránka umístěná na následující odkazu porovnává kompatibilitu jednotlivých CSS verzí a podporu jejich vlastností s vykreslovacími jádry různých webových prohlížečů.

http://en.wikipedia.org/wiki/Comparison_of_layout_engines_(CSS)

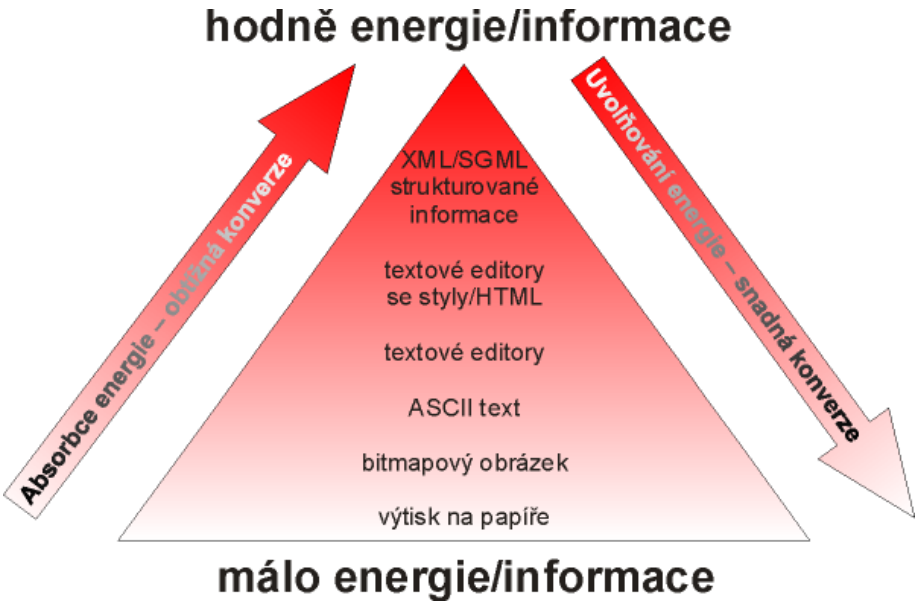
4 XML

XML je zkratka eXtensible Markup Language, česky může být přeložena jako rozšiřitelný značkovací jazyk. Tento jazyk vyvinulo konsorcium W3C k překonání nedostatků a omezujících prvků HTML. XML vzniklo, stejně jako HTML, z jazyka SGML (Standard Generalized Markup Language). Ve skutečnosti je XML tzv. ***metajazyk***, nadřazený značkovací jazyk, v rámci něhož je možné vytvářet vlastní jazyky (definované pomocí DTD popisu). Takovým jazykem je například XHTML, kombinace XML a HTML.

XML je formát dokumentů, který předepisuje, jak zapsat data společně s jejich významem. Díky tomu, že XML je jednoduchý formát, je rozšiřitelný (není omezen nějakou množinou elementů), je otevřený a je fakticky podporován (velkými i malými firmami), stal se z něho univerzální formát, který postupně nahrazuje dosud používané formáty. [34]

XML není jazyk pro zobrazení dokumentu (jako třeba HTML), ale pro popis dat a jejich strukturalizaci. Zobrazením souboru XML se zabývá úplně jiný jazyk: XSL (eXtensible Stylesheet Language), případně v předchozí kapitole zmiňované kaskádové styly CSS.

Velkou výhodou dokumentů psaných v XML je, že mohou být přemístěny do jakéhokoli formátu na jakékoli platformě aniž by použité elementy ztratily svůj význam. To znamená, že je možné využívat stejnou informaci v internetovém prohlížeči, PDA, nebo jakémkoli zařízení, které umí tuto informaci náležitě využít. [35]



Obrázek 22. Poměr množství informace v XML dokumentu k ostatním formátům

Zdroj: http://www.kosek.cz/clanky/xml/xml-01.png

4.1 Syntaxe

Každý dokument XML se skládá ze znaků, mezi kterými jsou uvedena data. Znaky v XML se nazývají ***elementy*** a zapisují se pomocí ***tagů*** a většině elementů odpovídají dva tagy – počáteční a ukončovací: **<element>**obsah elementu**</element>**. Pravidla pro zápis XML stanovují, že všechny elementy musí být párové, to znamená musí obsahovat tag počáteční i tag ukončovací. Pokud je v zápisu uveden pouze tag počáteční, hlásí program při zobrazení chybu a dokument nenačte. Jediná výjimka je, pokud je tag prázdný <element/>. V XML nejsou žádné předdefinované tagy, proto si autor může vytvořit právě takové tagy, jaké potřebuje. Například pro vyjádření jména v adresáři lze vytvořit následující tag:

```
<jmeno>Honza</jmeno>
```

Každý element může obsahovat kromě svého jména ještě další informace, které blíže specifikují jeho obsah. Těmto "informacím o informacích" se říká ***metadata*** a jsou ukládány v tzv. ***atributech***. Atributem u elementu <cena> by mohla být například měna, která udává, že cena je uvedena v korunách českých. Atributy se zapisují v počátečním tagu oddělené od názvu elementu mezerou. Jeden element může mít několik atributů, které se od sebe oddělují také mezerou. [36]

```
<cena měna="CZK">450.000</cena>
```

Názorná ukázka, jak by mohl vypadat XML dokument v praxi:

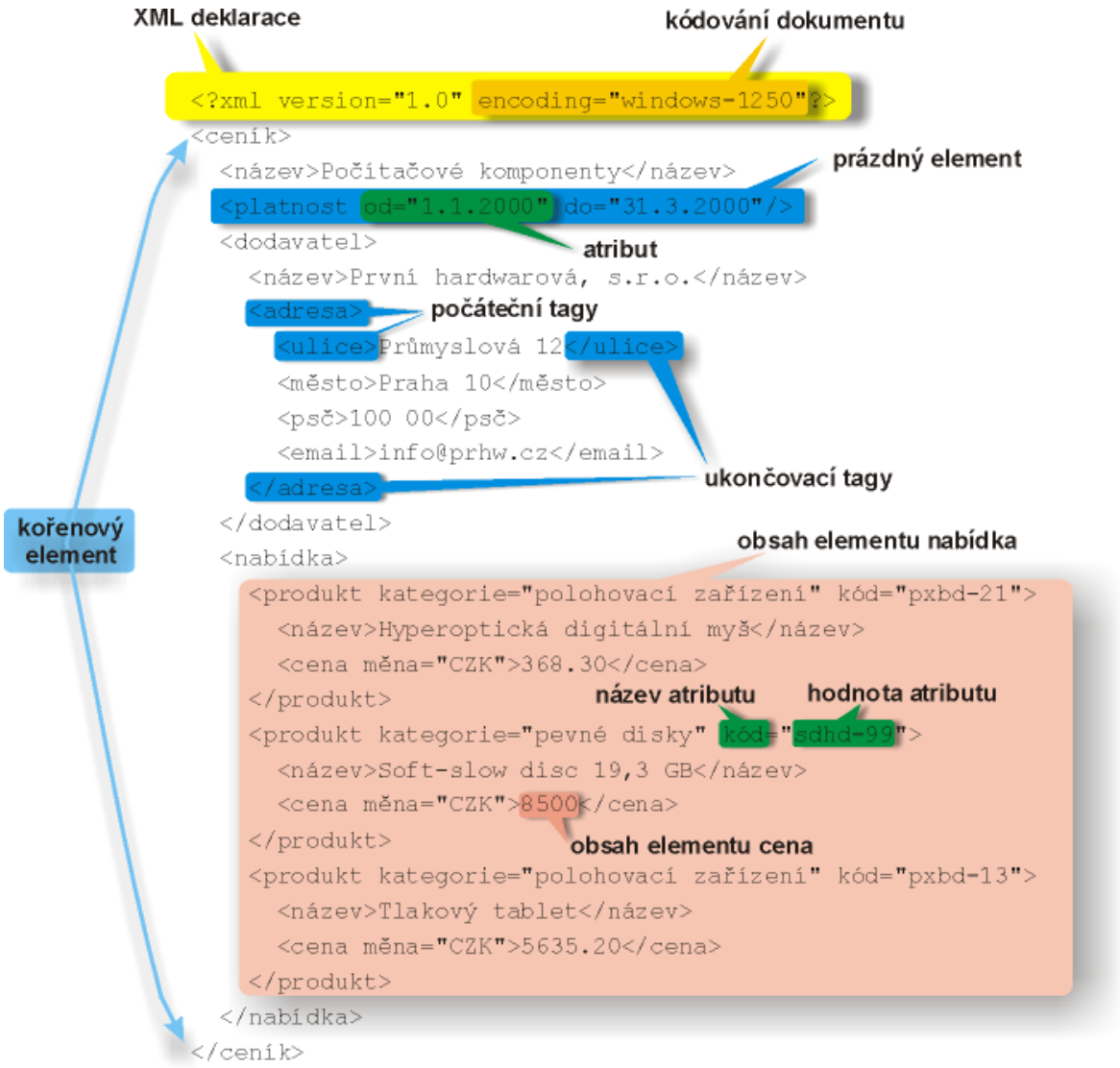
```
<? xml version="1.0" encoding="windows-1250"?>
<adresy>
   <osoba>
      <jmeno>Pepa</jmeno>
      <prijmeni>Mrak</prijmeni>
      <mesto>Praha</mesto>
   </osoba>
   <osoba>
      <jmeno>Jirka</jmeno>
      <prijmeni>Drak</prijmeni>
      <mesto />
   </osoba>
</adresy>
```

Prvnímu řádku se říká deklarace XML dokumentu, je zde uveden druh a verze jazyka (xml version="1.0") a způsob kódování dokumentu, který se určí pomocí parametru encoding (encoding="windows-1250").

Tag `<adresy>` je tag na nejvyšší úrovni, tzv. ***kořenový tag***. Každý dokument XML musí mít právě jeden tag na úrovni kořene. [37]

Elementy mohou být do sebe ***vnořovány***, ale nesmí dojít k jejich ***překřížení*** např.

```
<firma>yyy<adresa></firma></adresa>
```



Obrázek 23. Základní syntaxe jazyka XML

Zdroj: http://www.kosek.cz/clanky/swn-xml/syntaxe-zaklady.png

4.2 Názvy tagů v jazyce XML

Už bylo zmíněno, že v XML nejsou žádné předdefinované tagy. Existují jistá pravidla pro tvorbu jejich názvů. Názvy musí začínat buď písmenem nebo znakem podtržítka ("_"). Zbytek názvu může obsahovat písmena, číslice, podtržítko, tečku nebo pomlčku. Mezery jsou však vyloučeny. Názvy nemohou začínat řetězcem "xml", který je vyhrazen pro specifikace samotného jazyka XML.

V XML se rozlišují malá a velká písmena. Velmi často používaná konvence určuje, že se tagy píší malými písmeny, a pokud se tag skládá z více slov, píše se první písmeno každého slova velkým písmenem. Pro uvození celého jména jsou možnosti názvu tagu následující:

```
<celeJmeno> <CELE-JMENO> <Cele_Jmeno> <CELEJMENO> <celejmeno>
```

Je nutné si dát pozor na správný zápis názvů tagů. Pokud bude jako počáteční tag `<celejmeno>` a ukončovací `</celeJmeno>`, prohlížeč nahlásí chybu a dokument nezobrazí.

Dále se v názvech tagů nedoporučuje používat dvojtečku. Tento znak je vymezen pro vyjádření jmenných prostorů. [37]

4.3 Související technologie

Rozsah této práce nedovoluje podrobněji popsat následují způsoby práce s XML dokumenty, proto jsou zmíněny pouze okrajově.

4.3.1 DTD – definice typu dokumentu

Původně standard XML počítal pouze s jedním způsobem určení struktury dokumentů, a to DTD (Document Type Definition). XML schéma bylo navrženo teprve později. Řeší některé nedostatky DTD, jako například chybějící datové typy a menší možnosti restrikce hodnot, na druhou stranu je výrazně komplexnější a vazba na dokument XML není tak jednoduchá jako v případě DTD.

DTD obsahuje určení elementů, které se mohou v dokumentu vyskytovat, jejich atributů a obsahu, jež mohou elementy mít (tj. to, co se smí objevit mezi počátečním a ukončovacím tagem). V DTD je také možné definovat entity pro použití v dokumentu.

Deklarace elementů

Element je deklarován pomocí stejnojmenné instrukce:

```
<!ELEMENT jméno_elementu definice_obsahu_elementu>
```

Například tedy takto:

```
<!ELEMENT pokus (#PCDATA)>
```

Jako definici obsahu elementu lze použít klíčová slova `EMPTY` (prázdný element, jenž může obsahovat pouze atributy) a `ANY` (element bez jakéhokoli omezení obsahu). Pokud není zvolena žádná z těchto možností, je třeba v kulatých závorkách uvést ***model elementu***. Model elementu určuje, jaké elementy a v jakém pořadí se mohou uvnitř deklarovaného elementu objevit. [38]

Podrobnější informace k modelu elementu, tzn. další speciální klíčová slova a vyhrazené znaky, je možné získat v knize [38] na straně 14.

Deklarace atributů

Povolené atributy určuje DTD pomocí deklarace `ATTLIST`:

```
<!ATTLIST jméno_elementu jméno_atributu typ_atr. modifikátor>
```

V jedné deklaraci lze specifikovat všechny možné atributy elementu, stejně jako pro každý z nich použít samostatnou direktivu `ATTLIST`. Nejčastěji se jako typ atributu používá `CDATA`, tj. běžný znakový řetězec. Dále lze použít typy `ENTITY` (odkaz na externí entitu), `NMTOKEN` (identifikátor, který může začínat číslicí), `ID` (jednoznačný identifikátor v rámci dokumentu), `IDREF` (odkaz), `NOTATION` (výběr některé z dříve definovaných notací) či pouhý výčet hodnot.

```
<!ATTLIST osoba pohlaví (žena | muž)>
```

Pomocí direktiv `#REQUIRED` a `#IMPLIED` je dále možné určit, zda je daný atribut vyžadován nebo zda se předpokládá určitá standardní hodnota. [38]

4.3.2 XSD - XML schéma

Mezi zásadní výhody XSD (XML Schema Definition) oproti DTD patří (převzato z [38]):

- Podpora jmenných prostorů
- Definice struktury je též ve formátu XML a pro návrh schémat tedy lze použít běžné nástroje XML
- Schémata umožňují definovat datové typy a velmi přesně specifikovat omezení možných hodnot
- V případě násobnosti elementů lze kontrolovat možný počet opakování

Je třeba si uvědomit, že použití XML schémat má i své nevýhody, mezi něž lze zařadit:

- Podstatně složitější mechanizmus
- Slabší podpora v aplikacích
- Nemožnost určení kořenového elementu (v DTD pomocí hlavičky Doctype)
- Nelze rozdělit definici struktury na interní a externí část

Deklarace elementů

Základem schématu XML je hierarchická struktura odpovídající struktuře požadovaného dokumentu, tvořená převážně elementy `element`. Kořenovým elementem je vždy element `schema`, jenž celé schéma zastřešuje.

```
<xsd:element name="koren"> …  </xsd:element>
```

Výše uvedený fragment definuje element jménem `koren`, uvnitř elementu `element` je potom přesněji specifikován obsah daného elementu (model). V nejjednodušším případě tento element obsahuje pouze text. Pak stačí nastavit vlastnost type na hodnotu `string`:

```
<xsd:element name="koren" type="xsd:string" />
```

Pokud pouze textový element nestačí, je třeba určit typ pomocí vnořeného elementu `complexType`, jenž umožňuje vnořování elementů či atributů. Parametr `mixed` tohoto elementu určuje, zda má vytvářený element smíšený obsah (může kromě elementů obsahovat i text) či zda smí obsahovat pouze další, vnořené elementy. Obsah elementu `complexType` závisí na vytvářeném modelu.

Definice atributů

Atributy lze jednotlivým elementům přidat pomocí vnořeného elementu `attribute`:

```
<xsd:attribute name="titul" type="xsd:string" />
```

U atributů je možné určit, jak mohou být používány. Není-li určeno jinak, je každý definovaný atribut volitelný, což znamená, že ve zdrojovém dokumentu může a nemusí být uveden. Jeho použití lze vynutit přidáním atributu `use` s hodnotou `required`. Naopak hodnota `prohibited` se používá v situacích, kdy atribut ve zdrojovém kódu nesmí být uveden. Standardní hodnota atributu `use` je `optional`, tj. zmíněná volitelnost. V případě, že je atribut volitelný, je možné definovat jeho standardní hodnotu pomocí atributu `default`. [38]

Silnou stránkou XML schémat je možnost velmi přesně určit možné hodnoty elementů či atributů. K tomuto účelu existuje několik jednoduchých datových typů. Výčet těchto datových typů a další informace k práci s XML dokumenty je v knize [38].

V návaznosti na vznik jazyka XML začaly vznikat nové technologie. Mezi ty nejznámější patří: XHTML, XSLT pro transformace XML dokumentů, XPath pro navigaci v rámci XML dokumentů, DOM jako standard pro přístup a manipulaci XML dokumentů, SOAP pro vzdálenou komunikaci přes HTTP, SVG jako grafický vektorový formát a celá řada dalších. [39]

II. PRAKTICKÁ ČÁST

5 ZABEZPEČENÍ MICROSOFT WINDOWS XP PROFESSIONAL

V zásadě existují dvě hlavní skupiny hrozeb, kvůli kterým se zabezpečují operační systémy Windows (nejen ve verzi XP). V první skupině se nachází škodlivý software, který se snaží různými způsoby omezit funkčnost systému, poškodit uložená data nebo naopak tato data a jiné informace zpřístupnit třetí osobě. Takto škodlivý software se nazývá **malware** [40] a do počítače proniká většinou z Internetu. Pod souhrnné označení malware se zahrnují počítačové viry, internetoví červi, trojské koně, spyware a adware. Do druhé skupiny patří samotní uživatelé, kteří buď vědomě nebo nevědomě svým zásahem do systému (smazáním důležitých souborů, změnou údajů v registrech, instalací alternativních ovladačů, …) ohrozí stabilitu a bezpečnost systému.

Jedním z cílů této práce je ochránit počítač, resp. operační systém před druhou skupinou hrozeb, tzn. před destruktivními zásahy uživatelů. Nejúčinnější ochranou je v tomto případě omezení možností práce běžného uživatele se systémem pomocí nastavení uživatelských účtů a politiky účtů.

5.1 Uživatelské účty a skupiny zabezpečení

K této kapitole by se toho dalo napsat (a taky už bylo napsáno) mnoho, proto budou jen okrajově zmíněny základní pojmy týkající se této problematiky, které však budou vyváženy praktickou ukázkou zabezpečení systému.

Existuje nesčetně knih a internetových článků zabývajících se touto tématikou. Mezi kvalitní tituly přeložené do češtiny se řadí [41] [42].

Pro úlohy zabezpečení je důležité vytváření a odstraňování uživatelských účtů a definice a použití skupin zabezpečení. Definice bezpečnostních omezení nebo oprávnění, jež se mohou vztahovat k různým skupinám uživatelů a prostředků v síti, pomáhá zjednodušit implementaci a správu oprávnění a omezení, například v různých organizacích.

5.1.1 Uživatelské účty

Během instalace systému Windows XP Professional se automaticky vytvářejí dva uživatelské účty – ***Administrator*** (správce) a ***Guest*** (host). Účet Administrator lze používat k počátečnímu přihlášení a konfigurování počítače. Účet Guest je po instalaci automaticky zakázán. Jak již je zřejmé z názvů, uživatelské účty se dělí do několika skupin podle úrovně omezení jejich přístupu k systému:

- **účet správce** – uživatelský účet, který je členem skupiny Administrators. S tímto účtem jsou spojena úplná oprávnění k počítači a jeho řízení. Pomocí účtu správce lze získat přístup ke všem uživatelským účtům v počítači a upravovat je a také přidat nebo odebrat programy či hardwarová zařízení.
- **účet hosta** – účet, který má maximálně omezená práva. Používá se v případech, kdy je potřeba dovolit různým uživatelům, aby se přihlásili a využívali místních prostředků, aniž by bylo nutné vytvářet pro každého uživatele samostatné heslo. Jinak se doporučuje účet Guest nechat zakázaný.
- **uživatelský účet** – místní uživatelský účet musí vytvořit člen skupiny Administrators po skončení instalace systému. Účet je kolekce nastavených hodnot zahrnující jedinečné uživatelské jméno, heslo a sadu práv a oprávnění k používání prostředků počítače. [42]

Účet Administrator nelze odstranit, zakázat nebo odebrat z místní skupiny Administrators. Tím je zajištěno, že nikdy nemůže být zablokován přístup k počítači odstraněním nebo zákazem všech účtů správce. Touto funkcí se účet Administrator odlišuje od ostatních členů místní skupiny Administrators.

Pokud je systém Windows XP Professional spuštěn s účtem správce, je zranitelný trójskými koni a jinými rizikovými faktory. V takovém případě může trójský kůň stažený z Internetu vykonávat takové činnosti, jako je formátování disku, odstranění všech souborů, vytvoření nového uživatele s oprávněními správce apod. Proto se doporučuje vytvořit si vlastní účet a přidat ho do skupiny User nebo Power Users a snížit tak riziko poškození systému. [44]

Následující tabulka uvádí přehled oprávnění, která jsou k dispozici u zmíněných typů účtů. [43]

Tabulka 2: Přehled oprávnění pro různé typy účtů

Akce	Uživatel:	Uživatel se standardním oprávněním:	Uživatel s omezeným oprávněním:
	správce	[omezený účet]	[účet hosta]
Vytváření uživatelských účtů	ANO		
Změna systémových souborů	ANO		
Změna nastavení systému	ANO		
Čtení souborů jiných uživatelských účtů	ANO		
Přidání nebo odebrání hardwaru	ANO		
Změna hesel jiných uživatelských účtů	ANO		
Změna hesel vlastních uživatelských účtů	ANO	ANO	
Instalace libovolných programů	ANO		
Instalace většiny programů	ANO	ANO	
Ukládání dokumentů	ANO	ANO	ANO
Používání nainstalovaných programů	ANO	ANO	ANO

5.1.2 Skupiny zabezpečení

Uživatelské účty jsou rovněž členy skupin zabezpečení. Podle organizačního prostředí se ke správě zabezpečení používají skupiny, jež mohou být definovány podle rozsahu, účelu, práv a nebo rolí. Rozsah skupiny zabezpečení může zahrnovat jediný počítač, jednu doménu nebo několik domén v rámci doménové struktury. Obecně spadají skupiny systému Windows XP Professional do jedné z několika kategorií. V případě nastavení zásad zabezpečení pro počítač s dotykovým panelem nepřipojeným do sítě se využije kategorie **Místní skupiny zabezpečení v počítači**. Více o kategoriích skupin zabezpečení je v kapitole 16, "Ověřování a řízení přístupu" v knize [42].

Systém Windows XP Professional obsahuje celou řadu předdefinovaných místních skupin zabezpečení v počítači. Členství ve skupině poskytuje uživateli práva a možnosti provádět v počítači různé úkoly. Uživatelský účet může být členem jedné nebo více skupin.

Používá-li se systém Windows XP Professional v konfiguraci jako samostatný počítač nebo v rámci pracovní skupiny, lze pomocí panelu *Uživatelské účty*, v okně *Ovládací panely* spravovat pouze tři z těchto předdefinovaných skupin zabezpečení – skupinu ***Administrators***, ***Users*** a ***Guests***. Používá-li se systém Windows XP Professional v konfiguraci jako samostatný počítač nebo v rámci pracovní skupiny, a je-li třeba používat také další skupiny zabezpečení, jejich správa se provádí z modulu snap-in *Místní uživatelé a skupiny*. [42]

Názvy jednotlivých skupin zabezpečení místního počítače a s nimi spojené role a oprávnění jsou následující: [44]

- **Administrators –** členové skupiny Administrators mají největší možnosti výchozích oprávnění, která jim umožňují úplnou kontrolu nad celým systémem. Mohou také měnit vlastní oprávnění. Úkolem členů skupiny Administrators je provádění úkolů souvisejících s údržbou počítače.
- **Power Users –** členové skupiny Power Users mohou vytvářet uživatelské účty, ale upravit a odstranit mohou pouze účty, které vytvořili. Mohou také odebírat uživatele ze skupin Power Users, Users a Guests. Nemohou přebírat vlastnictví souborů, zálohovat nebo obnovovat adresáře, zavádět ovladače zařízení.
- **Users –** skupina Users se vyznačuje nejvyšší úrovní zabezpečení, neboť výchozí oprávnění přidělená této skupině nepovolují členům měnit nastavení operačního systému ani jiná uživatelská data. Členové skupiny Users mohou provádět většinu běžných úkolů, například spouštět aplikace, používat místní a síťové tiskárny a vypínat a zamykat pracovní stanice.
- **Guests –** skupina Guests umožňuje příležitostným nebo jednorázovým uživatelům přihlásit se k účtu Guest a získat omezené možnosti. Členové skupiny Guests mohou také vypnout systém v pracovní stanici.
- **Backup Operators –** členové této skupiny mohou zálohovat a obnovovat soubory bez ohledu na oprávnění, která tyto soubory chrání. Mohou se přihlásit do počítače a vypnout jej, ale nemohou změnit nastavení zabezpečení.
- **Replicator –** její členové mají oprávnění k replikaci souborů v rámci domény.
- **Repote Desktop Users –** členové mají oprávnění ke vzdálenému přihlašování

5.2 Provedená nastavení

Z důvodu zabezpečení počítače, který nebude pod stálým dohledem a bude k němu mít přístup větší množství uživatelů, bylo zapotřebí vytvořit účet s omezenými právy. Běžný uživatel nebude mít k dispozici klávesnici ani myš, čímž jsou jeho možnosti zasahovat do systému značně omezené, i když pořád ne zanedbatelné.

Omezení přístupu do systému přes uživatelský účet jde několika způsoby. Prvním z nich je zásah do registrů systému, dále pak nastavením přes *Místní zásady zabezpečení* nebo *Správa počítače –> Místní uživatelé a skupiny*. Další možností je použití konfiguračního nástroje "Local Group Policy Editor", jež lze spustit přes *Start –> Spustit…* –> zadáním příkazu ***gpedit.msc***. Případně lze použít malé programy (utility), které umožňují při troše cviku snadně a rychle vytvořit zabezpečený uživatelský účet.

Pro zabezpečení počítače byly zvoleny 2 aplikace:

- **Windows SteadyState 2.0** – dřívější název Shared Computer Toolkit, se snaží o dosažení větší spolehlivosti sdílených počítačů v práci, na veřejných místech nebo školách a tím správcům ušetří čas strávený údržbou. Umožňuje spravovat uživatelské účty a nastavovat omezení pro práci uživatele se systémem (odebráním ikon z plochy, zakázáním pravého tlačítka myši, zredukováním nabídky Start na potřebné minimum, atd.).
- **Tweak UI 2.10** – je z rodiny programů Microsoft PowerToys a umožňuje nastavit mnoho skrytých možností pro nastavení uživatelského rozhraní, které nejsou přístupné žádným jiným způsobem. Program pracuje dobře pouze ve Windows XP s nainstalovaným SP1 a vyšší a ve Windows Server 2003.

Obě tyto aplikace jsou zdarma ke stažení na stránkách Microsoftu pro uživatele legálního systému Windows. Způsob jejich nastavení je uveden v přílohách P I – P VIII.

6 KIOSKOVÝ MÓD V OPEŘE

Pro správnou funkci rozhraní, vytvořeného v Adobe Flex, je nutné ho spouštět přes webový prohlížeč na lokálním serveru (tzv. localhost). Na výběr byl trojlístek nejznámějších a nejpoužívanějších prohlížečů na platformě Windows – Internet Explorer 7, Firefox 2.0.0.14 a Opera 9.27. Důležitým kritériem pro výběr toho správného byla existence kioskového módu, tzn. prohlížeč pracuje v režimu celé obrazovky (full-screen). Takový mód znemožní běžnému uživateli opuštění okna prohlížeče, přístup do nastavení, ukončení aplikace a také jakékoliv zasahování do systému. Při pokusu uzavřít okno prohlížeče se automaticky otevře nové s nastavenou výchozí stránkou. Na tu se také prohlížeč vrátí po předem definované době nečinnosti.

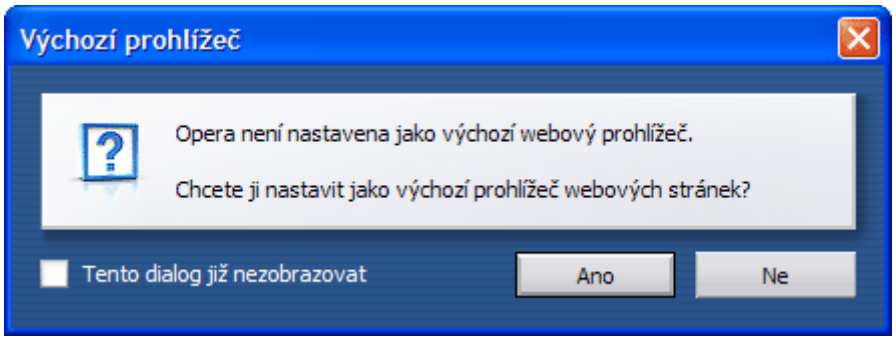
Všechny tyto prohlížeče nabízí zobrazování obsahu v celoobrazovém módu. Nakonec byl zvolen prohlížeč Opera, který již v základu nabízí široké možnosti nastavení (přizpůsobení ovládacího panelu s tlačítky, filtrování adres na základě URL, omezení nebo zakázání klávesových zkratek, v případě Opery také gesta myší, aj.). Například prohlížeč Firefox tyto funkce nabízí až po doinstalování doplňku (add-on) s názvem Open Kiosk 2.13, nebo R-kiosk 0.7.0.

Obrázek 24: Loga prohlížečů Firefox, Internet Explorer a Opera

Zdroj: http://www.noscope.com/media/opera_logo_comparisons.jpg

6.1 Opera jako výchozí prohlížeč

První možností jak Operu nastavit jako výchozí prohlížeč je potvrdit dialogové okno při prvním spuštění prohlížeče po instalaci. Dialogové okno vypadá následovně:



Obrázek 25: Nastavení Opery jako výchozího prohlížeče při prvním spuštění

Další možností je dodatečné nastavení, které se provede v menu Opery: *Nástroje –> Pokročilé volby –> Programy*, u volby *Při spuštění Opery ověřit, je-li výchozím prohlížečem* kliknout na tlačítko "Podrobnosti…". Zde se nastaví asociace s dokumenty HTML, adresami URL a protokoly HTTP, HTTPS a FTP (podle potřeby i dalšími). Při následném spuštění Opery se zobrazí dotaz, zda se má Opera nastavit jako výchozí prohlížeč – stačí potvrdit kliknutím na "Ano".

Když se potom Internet Explorer zeptá, zda ho nastavit jako výchozí, tak se musí kliknout na "Ne". Mimo to je možné IE přinutit, aby si nekontroloval, jestli je výchozí. Vypne se to pomocí: *Start –> Ovládací panely –> Možnosti internetu –> Programy –>* zrušit zatržení u položky *Aplikace Internet Explorer by měla zkontrolovat, zda je nastavena jako výchozí prohlížeč*.

6.2 Nastavení Opery v kioskovém módu

Opera nabízí široké možnosti zabezpečení prohlížeče (resp. celého systému) při spuštění v kioskovém módu. Vedle standardního nastavení byly navíc provedeny úpravy pro zvýšení zabezpečení. Jako zdroj informací posloužila online nápověda Opery, která se nachází na adrese [45].

6.2.1 Základní nastavení

S uvedeným nastavením se Opera automaticky spouští v kioskovém módu:

- Celoobrazový mód je povolený
- Panely nástrojů jsou zakázány
- Menu prohlížeče je zakázáno
- Všechny panely zůstávají přístupné (přes upřesňující parametry – viz. Pokročilé nastavení)
- Otevírání nových stránek v popředí i pozadí je zakázáno
- Uživatel nemůže tento mód opustit stisknutím klávesy Esc
- Vstup do systému a ostatních programů je zakázán blokováním klávesových zkratek Ctrl+Esc, Alt+Tab a Alt+Escape
- Potvrzovací tlačítko je zakázáno z důvodu zabránění nahrání souboru
- Nápověda není přístupná
- Pokud uživatel uzavře stránku, automaticky se načte domovská stránka a okno se maximalizuje

6.2.2 Pokročilé nastavení

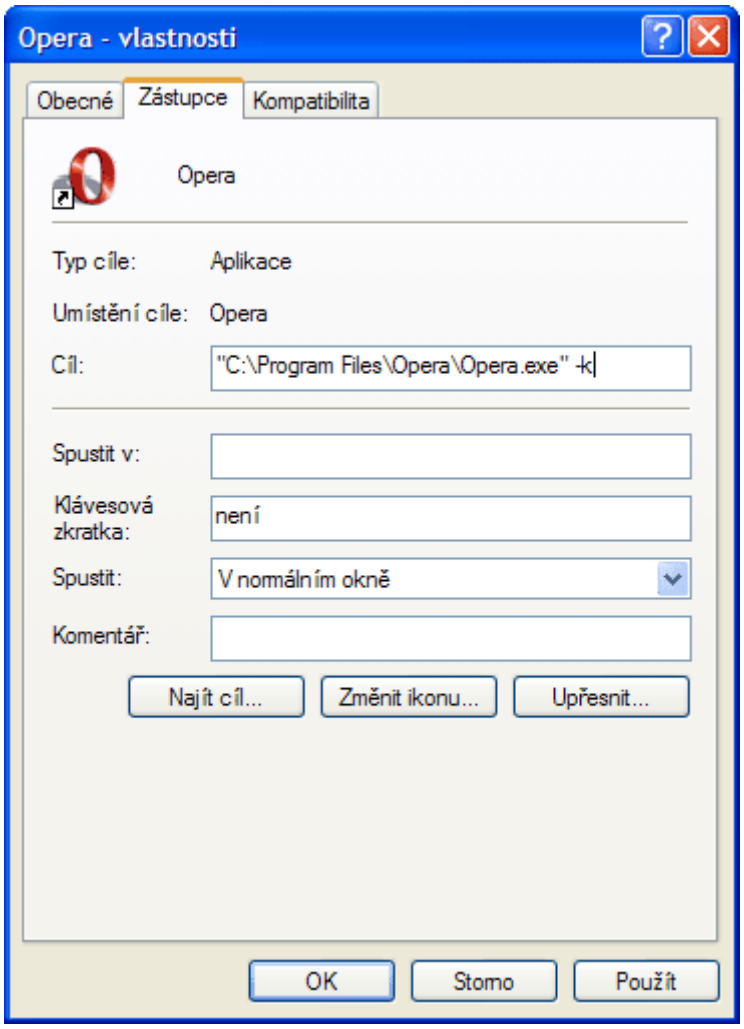
Pokročilé nastavení se provádí pomocí zadaných parametrů při spuštění. Pro spuštění Opery v kioskovém módu je třeba ji spustit s parametrem *k* nebo *kioskmode*.

Parametry, se kterými se Opera spouští automaticky:

- nochangebuttons – lišta s navigačními tlačítky se nezobrazí
- nochangefullscreen – nelze přepínat mezi celoobrazovým módem a klasickým zobrazením
- nosysmenu – odstraní menu prohlížeče (Soubor, Nástroje, Záložky, …)

Pro zlepšení zabezpečení a zvýšení komfortu práce s prohlížečem byly přidány následující parametry:

- kioskbuttons – zpřístupní lištu s tlačítky a lištu průběhu (ta umožňuje zobrazení adresy a tlačítek Dopředu, Zpět, Znovu načtení, atd.)
- kioskwindows – zobrazí lištu listů s aktuálně otevřenými stránkami, mezi kterými se lze snadno orientovat
- nodownload – dialog pro stahování je zakázán
- nokeys – zablokuje klávesové zkratky
- nomaillinks - zablokuje "mailto:" odkazy, tzn. nespustí se žádný e-mailový klient



Obrázek 26: Vložení parametrů pro spuštění v kioskovém módu

V klasickém zobrazení je v případě potřeby možné upravit lišty, které jsou dodatečně zobrazeny v kioskovém módu pomocí parametrů(v kioskovém módu není povolena editace lišt). Je možné přidat tlačítko domovské stránky, odebrat tlačítka pro tisk, aj.

Další možnosti nastavení prohlížeče pomocí parametrů jsou uvedeny na stránce [46].

6.2.3 Filtrování adres

Filtrování slouží k omezení přístupu prohlížeče na určité internetové stránky, nebo zakázání stahování určitých typů souborů. Nastavení filtru se provádí v konfiguračním prostředí prohlížeče zadáním "opera:config" do adresového řádku. Na nově otevřené stránce se zvolí záložka *Network*, ve které se nachází položka *URL Filter File*. Zde je uvedený odkaz na soubor, který obsahuje definici povolených URL.

Filtr je uložen v klasickém ini souboru. Může být rozdělen na [include] a [exclude] část, neboli část, kde jsou uvedené povolené adresy (include) a část, která obsahuje výčet zakázaných URL adres (exclude). Filtr podporuje zástupné znaky * (nahrazuje libovolný text libovolné délky) a ? (nahrazuje jeden znak).

URL adresy, které nejsou výslovně povoleny, jsou automaticky zakázány. Seznam zakázaných adres má přednost před seznamem povolených adres. Prioritu seznamů je možno prohodit.

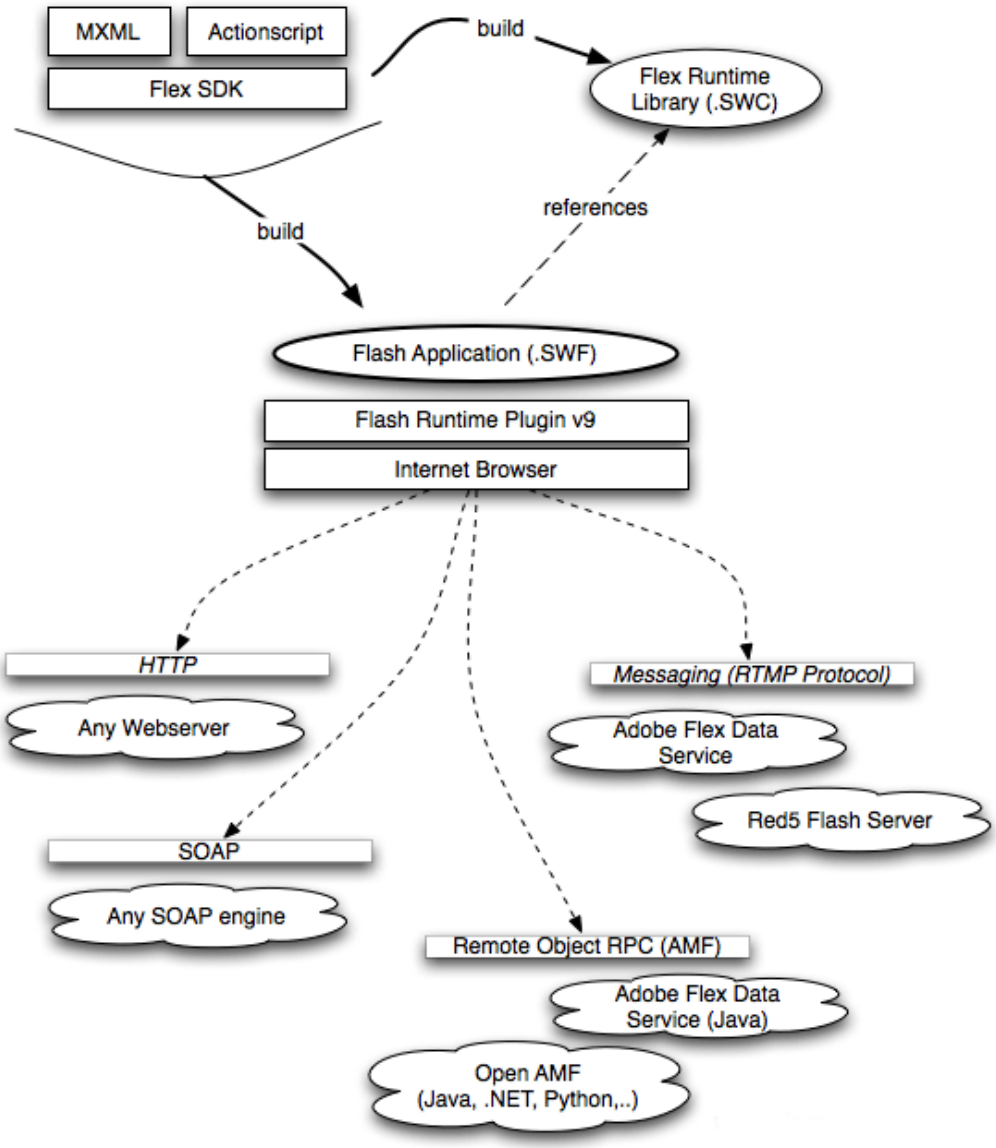
Postup jak nastavit filtrování URL v Opeře je na stránce [45].

V ukázce možného nastavení filtru jsou zakázány lokální soubory (protože nejsou povoleny) a je blokováno načítání obrázků ve formátu bmp, gif přes ftp.

```
[prefs]
prioritize excludelist=1
[include]
http://*
ftp://*
[exclude]
ftp://*.bmp
ftp://*.gif
ftp://*.jpg
```

7 ADOBE FLEX

Adobe Flex je kompletní řešení pro tvorbu uživatelsky bohatých aplikací (RIA) pro web a desktop. Do tohoto řešení spadá knihovna komponent a tříd na tvorbu uživatelského rozhraní (UI) a práci s daty – **Flex Framework**, integrované vývojářské prostředí (IDE) – **Flex Builder** a v neposlední řadě prostředí pro spouštění a běh RIA aplikací – **Flash Player** a **AIR**. Výhoda Flex aplikací spočívá v tom, že je lze jednoduše portovat na web (Flash Player) nebo desktop (AIR) bez nutnosti vyvíjet aplikaci dvakrát. Flex je kompatibilní se všemi HTTP servery a server-side programovacími jazyky. [47]



Obrázek 27. Možnosti Flex architektury

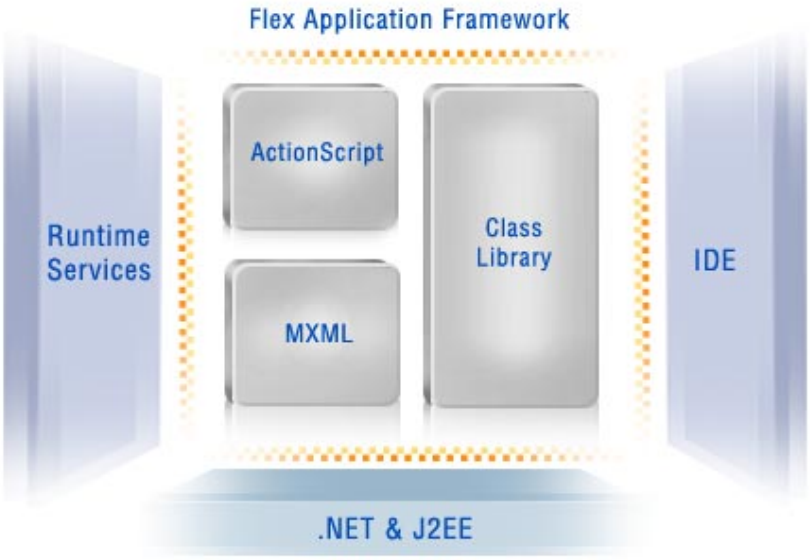
Zdroj: http://www.pierocampanelli.info/wp-content/uploads/2008/1/Flash.png

7.1 Součásti Flex 2

Produktová řada Flex 2 poskytuje novou generaci vývojářských nástrojů a služeb a skládá se z následujících částí:

- **ActionScript 3.0** – je plnohodnotný objektově orientovaný programovací jazyk, který stále posouvá možnosti Flash platformy. ActionScript 3.0 je navržen jako ideální jazyk k rychlé a efektivní tvorbě RIA. Tato nová verze se velmi rozšířila, zvýšila rychlost a zjednodušila vývoj komplexnějších aplikací. ActionScript je postavený na moderních standardech ECMAScript 4 a plně podporuje ECMA XML skriptovací standard E4X.
- **Flash Player 9** – jeho nová generace se výrazně zlepšila především v rychlosti zpracování kódu. K usnadnění těchto vylepšení obsahuje úplně novou, vysoce optimalizovanou ActionScript Virtual Machine (AVM), pojmenovanou AVM2. Nová Virtual Machine je několikanásobně rychlejší, podporuje "run-time error reporting" a ohromně vylepšila debugging. Flash Player 9 současně obsahuje AVM1, který zpracovává kód ActionScriptu 1.0 i 2.0, a je tak zpětně plně kompatibilní s existujícím obsahem.
- **Flex Framework 2** – přidává k základním částem FP9 a ActionScriptu 3.0 bohatou knihovnu tříd, která uživatelům umožňuje jednoduché využití nejlepších postupů v tvorbě úspěšných RIA. Flex používá jazyk založený na XML (pojmenovaný ***MXML***) a dovoluje vývojářům deklarativním způsobem stavět jednotlivé části aplikace. Vývojáři mají přístup k Flex Frameworku přes Flex Builder nebo přes Flex SDK, který je zdarma a obsahuje on-line kompiler a debugger, dovolující vývojářům použít jakýkoliv jimi oblíbený editor.
- **Flex Builder 2** – je nové integrované vývojářské prostředí (IDE) navržené se záměrem poskytnout vývojářům profesionální nástroj přímo pro tvorbu RIA. Stojí na standardu v odvětví open-source vývojovém prostředí ***Eclipse***. Flex Builder poskytuje vynikající programovací a ladící prostředí (obsahuje kompiler a debugger). Další výhodou platformy Eclipse spočívá v poskytování bohaté možnosti rozšíření, takže si lze IDE přizpůsobit svým vlastním potřebám a preferencím.

- **Flex Data Services 2** – rozšiřují klientský Flex Framework a poskytují výkonné propojení s existující serverovou logikou a daty. Přináší zároveň i služby k automatické synchronizaci dat mezi klientem a serverem, přidávají podporu ***push*** a ***publish/subscribe messaging*** a umožňují vyvíjet sdílené aplikace.[15]



Obrázek 28. Flex Application Framework

Zdroj: http://www.webspiders.com/en/images/flex_application_framework.jpg

7.2 Flex Builder 2

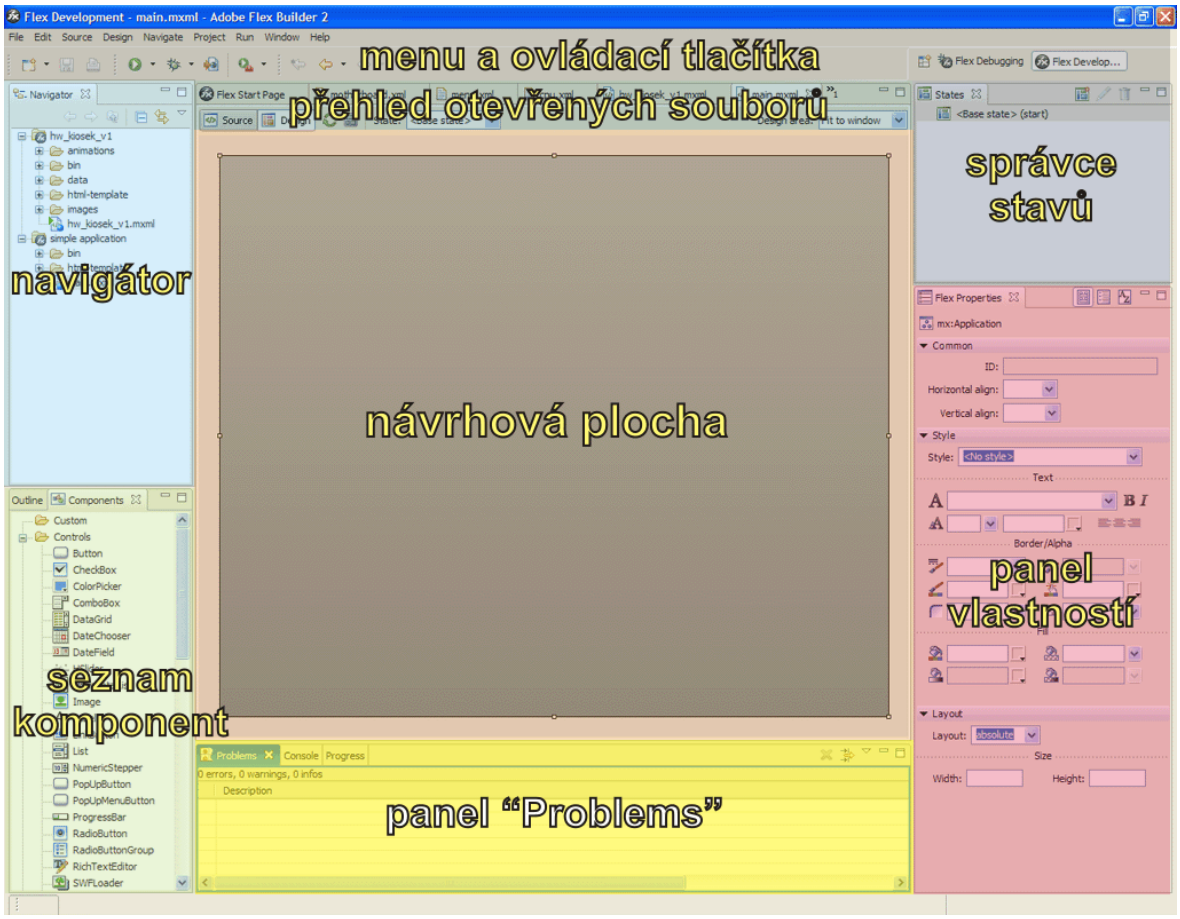
Flex Builder 2 je vývojové prostředí pro tvorbu Flex aplikací, které bylo vydáno v roce 2006. Samotné prostředí Flex Builderu je rozdělené do několika částí (panelů), které jsou přehledně uspořádané a programátor se v něm za chvíli dokonale orientuje.

Základní panely a ovládací prvky:

- **Navigátor** – zobrazuje veškeré dokumenty daného projektu.
- **Seznam komponent** (v grafickém módu) – obsahuje okno s připravenými komponentami nebo okno **Outline** (při práci se zdrojovým kódem), které slouží k přehlednému zobrazení objektů (elementů) použitých v aplikaci. Při editaci zdrojového kódu lze navíc přepínat mezi "MXML" zobrazením a "Class" zobrazením.

- **Hlavní část** – slouží jako návrhová plocha, která nabízí kompletní grafický náhled vyvíjené aplikace (v grafickém módu), nebo jako editor zdrojového kódu. Záložka "Design" slouží pro zobrazení grafického vzhledu aplikace a záložka "Source" slouží pro přepnutí do editoru zdrojového kódu.
- **Panel "Problems"** – nachází se ve spodní části a jeho úkolem je informovat o případných nesrovnalostech ve zdrojovém kódu aplikace.
- **Správce stavů** – seznam jednotlivých stavů objektů použitých v aplikaci.
- **Panel vlastností** – umožňuje definovat vzhled jednotlivých objektů, které jsou umístěné na scéně.

Umístění jednotlivých panelů na pracovní ploše je na následujícím obrázku.



Obrázek 29: Uspořádání panelů v aplikaci Adobe Flex 2

7.3 Základy tvorby Flex aplikací – ukázka programu

V této kapitole je ukázka mini aplikace vytvořené v prostředí Flex. Je v ní nastíněn způsob zápisu kódu v jazyce MXML, způsob volání funkcí napsaných v jazyce ActionScript a připojení externího CSS souboru s deklarací kaskádových stylů.

Po založení nového projektu přes *File –> New –> Flex project –> Basic* se automaticky vytvoří kostra aplikace a je možné okamžitě se přepnout se do grafického módu (tlačítko Design) a vkládat komponenty na pracovní plochu. Až jsou všechny potřebné komponenty na svých místech, přichází na řadu programátorská část. Tlačítkem "Source" se v hlavní části zobrazí zdrojový kód, včetně definic komponent vložených v grafickém módu.

Veškeré objekty, umísťované na scénu, jsou v MXML dokumentu definované jako jednotlivé elementy se specifickými atributy. Samotné zdrojové soubory jsou ukládány s koncovkou .mxml.

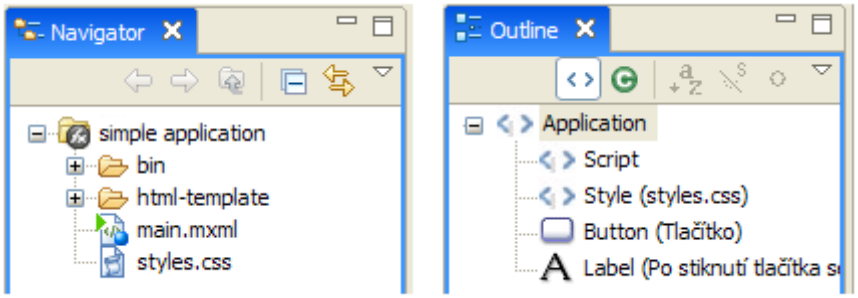
Při editaci zdrojového kódu je možné přiřadit tlačítku funkci "OnClick()", nadeklarovat tuto funkci pomocí ActionScriptu, případně uvést cestu k externímu CSS souboru (jako je tomu v ukázce).

MXML kód:

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <mx:Application xmlns:mx="http://www.adobe.com/2006/mxml"
3      layout="absolute" width="400" height="300" styleName="mainWindow">
4      <!--  -->
5      <mx:Script>
6          <![CDATA[
7              import mx.controls.Alert;
8              private function OnClick():void
9              {
10                 mx.controls.Alert.show('Tlačítko bylo stisknuto');
11             }
12         ]]>
13     </mx:Script>
14     <!--  -->
15     <mx:Style source="styles.css" />
16
17     <mx:Button label="Tlačítko" click="OnClick()" x="153" y="161"/>
18     <mx:Label text="Po stiknutí tlačítka se zobrazí popup okno"
19         styleName="simpleText" x="58.5" y="80"/>
20
21 </mx:Application>
```

CSS deklarace:

```
1  global {
2      fontFamily: Arial; fontSize: 18;}
3
4  .mainWindow {
5      borderStyle: solid; borderThickness: 3;
6      borderColor: #000000;}
7
8  Alert {
9      fontSize: 12;}
10
11 .simpleText {
12     fontWeight: bold; fontSize: 14;
13     fontStyle: italic;  color: #FFFFFF;}
```



Obrázek 30: Panely Navigátor a Outline ukázkové aplikace

Na dalším obrázku jsou zobrazena výsledná okna aplikace.

Obrázek 31: Výsledný vzhled ukázkové aplikace

Výsledná aplikace je uložena v souboru .swf.

7.4 Vytvořené rozhraní pro informační kiosek

Zatím nebylo nikde zmíněno, proč byl vybrán právě Flex pro tvorbu aplikace sloužící k zobrazování informací o hardwaru. Vedle Flexu bylo zvažováno i nasazení technologie Silverlight.

Mezi hlavní argumenty upřednostňující volbu Flexu patří:

- Flex framework a základní vývojářské nástroje jsou k dispozici zdarma (ve vývojářském kitu Flex SDK).
- Rozšířenost plug-inu Flash Player (má ho nainstalovaný až 90% prohlížečů).
- Flex je "léty prověřená technologie" a má za sebou mnoho let vývoje. V dubnu 2008 byla vydána již třetí verze vývojového prostředí Flex Builder 3.
- Z předchozího vyplývá, že existuje široká komunita vývojářů, kteří dokáží poradit začátečníkům. Také jsou k dispozici zdrojové kódy různých aplikací, které urychlují začínajícím programátorům proces učení.

Článek [48] potvrdil, že tento výběr byl správný.

Obrázek 32. Vytvořené rozhraní pro dotykový displej

ZÁVĚR

Účelem této práce bylo vytvořit rozhraní pro prezentaci hardwarových technologií, jejichž znalost by v dnešní době měla patřit mezi základní dovednosti každého, kdo přichází do styku s počítačem, potažmo s informačními technologiemi. Rozhraní obsahuje pouze základní přehled jednotlivých hardwarových komponent s přehledem jejich vývoje. Velkou výhodou je možnost neustálého doplňování nových informací k technologiím, které se neustále vyvíjejí kupředu.

Rozhraní je vytvořeno v aplikaci Adobe Flex, která nabízí ideální prostředí pro tvorbu bohatých internetových alikací (RIA). Možnosti takových aplikací jsou opravdu široké, jejich využitím a různými technologiemi na tvorbu RIA aplikací se zabývá druhá kapitola. Většina zobrazovaných informací je načítána z externích XML souborů, a proto je editace informací, v nich uložených, velmi snadná. V teoretické části je jedna kapitola věnována právě základům jazyka XML.

Vedle tvorby rozhraní se práce zabývá zabezpečením operačního systému Windows XP a internetového prohlížeče Opera před nepovolanými zásahy, které by mohly způsobit nefunkčnost celého informačního kiosku. Prohlížeč Opera je zabezpečen tím způsobem, že je spuštěný v tzv. kioskovém módu.

Přílohou práce je CD disk, který obsahuje kompletní diplomovou práci ve formátu Microsoft Word 2003 a ve formátu PDF. Na CD je rovněž rozhraní pro zobrazování informací o hardwaru. Také se zde nachází časově omezená verze programu Adobe Flex Builder 2 umožňující tvorbu flex aplikací.

Doufám, že výsledek mé práce bude prospěšný všem, kteří si chtějí prohloubit své znalosti v této tak rozsáhlé, leč velmi zajímavé problematice.

SEZNAM POUŽITÉ LITERATURY

[1] FRANKLIN, Derek. Macromedia Flash MX : Kompletní průvodce. 1. vyd. Brno : Computer Press, 2003. 846 s., 1 CD-ROM. ISBN 80-7226-831-7.

[2] KERMAN, Phillip. ActionScript ve Flashi : Podrobná příručka. 1. vyd. Praha : Computer Press, 2002. 507 s. ISBN 80-7226-615-2.

[3] Digital Media s.r.o.. Flash.cz - server pro kreativní lidi : Server o programech Flash, Dreamweaver, Fireworks, Photoshop [online]. c2005-2007 [cit. 2008-01-24]. Dostupný z WWW: <http://www.flash.cz/>.

[4] Adobe Systems Incorporated.. Produkty Adobe [online]. c2008 [cit. 2008-01-24]. Dostupný z WWW: <http://www.adobe.com/cz/>.

[5] Microsoft Corporation.. Microsoft Česká republika [online]. c2007 [cit. 2008-01-24]. Dostupný z WWW: <http://www.microsoft.com/cs/cz/>.

[6] Adobe Systems Inc.. Flex.org : Rich Internet Applications, Rapid Web Application Development, Open Source Flex, Open Source Flash, Adobe Flex, Flex 2, Flex 3 [online]. [2008] [cit. 2008-01-24]. Dostupný z WWW: <http://flex.org/>.

[7] GREPL, Zbyněk. Informační kiosky : přínosy nasazení [online]. 2004 [cit. 2008-03-23]. Dostupný z WWW: http://extranet.nmnm.cz/inovomestsko/prezentace.pdf

[8] VOJÁČEK, Antonín . Princip SAW dotykových ploch a displejů [online]. 1997-2005 [cit. 2008-03-25]. Dostupný z WWW: <http://automatizace.hw.cz/princip-saw-dotykovych-ploch-displeju>.

[9] How Does a Touchscreen Work? [online]. [2006] [cit. 2008-03-24]. Dostupný z WWW: <http://touchscreens.com/intro-anatomy.html>.

[10] Touchscreeny [online]. c2007 [cit. 2008-03-25]. Dostupný z WWW: <http://www.power.cz/page/122_touchscreeny/>.

[11] Compare All Resistive Touch Technologies [online]. c2008 [cit. 2008-03-22]. Dostupný z WWW: <http://www.elotouch.com/Technologies/compare_resist.asp>.

[12] Touchscreen Technology : Questions & Answers [online]. [2007] [cit. 2008-03-23]. Dostupný z WWW: <http://www.i-techcompany.com/touchscreen.html>.

[13] Surface Capacitive Touch Screen [online]. 2005-2008 [cit. 2008-03-26]. Dostupný z WWW: <http://www.amsimpex.com/products/capacitive-PCT-touchscreen.html>.

[14] SNÁŠEL, Jaroslav. Už vím, jak fungují dotykové displeje [online]. 4.11.2004 [cit. 2008-03-22]. Dostupný z WWW: <http://www.mobilmania.cz/default.aspx?article=1108570>.

[15] VESELKA, Aleš. Obohaťte své uživatele pomocí RIA [online]. 2007-02-05 [cit. 2008-04-04]. Dostupný z WWW: <http://www.symbio.cz/clanky/obohatte-sve-uzivatele-pomoci-ria.html>.

[16] PICHLÍK, Roman. Rich Internet Application [online]. 2005-06-14 [cit. 2008-04-05]. Dostupný z WWW: <http://interval.cz/clanky/rich-internet-application/>.

[17] TOTH, David. Co je to Ajax? [online]. 2007 [cit. 2008-03-26]. Dostupný z WWW: <http://webing.felk.cvut.cz/hs/download/DT-ajax-CZ-art.pdf>.

[18] Slovník internetových výrazů : AJAX [online]. c1999-2008 [cit. 2008-03-27]. Dostupný z WWW: <http://www.symbio.cz/slovnik/ajax.html>.

[19] Flash [online]. c2005-2008 [cit. 2008-03-27]. Dostupný z WWW: <http://www.adaptic.cz/znalosti/slovnicek/flash.htm>.

[20] DEB, Brijesh. Rich Internet Applications : A Look Into Available Technology Choices [online]. c2008 [cit. 2008-03-28]. Dostupný z WWW: <http://www.jaxmag.com/itr/online_artikel/psecom,id,828,nodeid,147.html>.

[21] ČÍŽEK, Jakub. Adobe AIR Beta 3: WEB 2.0 přichází na desktop [online]. 2008-01-07 [cit. 2008-04-07]. Dostupný z WWW: <http://www.zive.cz/default.aspx?server=1&article=139665>.

[22] BRICHTA, Ondřej. Flex, AIR novinky pro rok 2008 [online]. 2008-01-21 [cit. 2008-04-06]. Dostupný z WWW: http://www.flash.cz/portal/clanek.aspx?id=1049>.

[23] CHALUPA, Pavel. Co je to OpenLaszlo a k čemu je to dobré? [online]. 2006-05-24 [cit. 2008-03-27]. Dostupný z WWW: <http://www.root.cz/zpravicky/co-je-to-openlaszlo-a-k-cemu-je-to-dobre/>.

[24] OpenLaszlo Architecture : Overview [online]. 2002-2005 [cit. 2008-04-07]. Dostupný z WWW: <http://www.openlaszlo.org/lps/docs/guide/architecture.html>

[25] RIA - Rich Internet Application [online]. 2007-06-09 [cit. 2008-03-28]. Dostupný z WWW: <http://srakyi.modry.cz/blog/index.php?s=flexu>.

[26] JANOŠÍK, Dušan. Něco málo k technologiím WPF a Silverlight [online]. 2007-05-13 [cit. 2008-03-28]. Dostupný z WWW: <http://www.vyvojar.cz/Articles/473-neco-malo-k-technologiim-wpf-a-silverlight.aspx>.

[27] KŘÍŽ, Lukáš. XAML jako spasitel [online]. c2006 [cit. 2008-03-28]. Dostupný z WWW: <http://archiv.computerworld.cz/cwarchiv.nsf/clanky/DC4E932E3FBB4381C12571AA00470ED2?OpenDocument>.

[28] VÁVRů, Vlastimil. Pár střípků o SilverLight [online]. 2007-09-30 [cit. 2008-03-29]. Dostupný z WWW: <http://vavru.cz/ostatni/par-stripku-o-silverlight/>.

[29] MALÝ, Martin. Microsoft Silverlight - další do ringu k Adobe Flex [online]. 2007-05-03 [cit. 2008-03-28]. Dostupný z WWW: <http://dev20.info/microsoft-silverlight-dal-i-do-ringu-k-adobe-flex>.

[30] Silverlight Architecture [online]. Microsoft Corporation, c2008 [cit. 2008-04-12]. Dostupný z WWW: <http://msdn2.microsoft.com/en-us/library/bb404713.aspx>.

[31] BOS, Bert. Cascading Style Sheets home page [online]. 2008-04-10 [cit. 2008-04-14]. Dostupný z WWW: <http://www.w3.org/Style/CSS/>.

[32] CSS kaskádové styly [online]. c2004 [cit. 2008-04-13]. Dostupný z WWW: <http://www.webtvorba.cz/css/>.

[33] JANOVSKÝ, Dušan. CSS : Kaskádové styly [online]. 2008-01-14 [cit. 2008-04-13]. Dostupný z WWW: <http://www.jakpsatweb.cz/css/>.

[34] KOČÍ, Michal. Co je XML? [online]. 2000-02-21 [cit. 2008-04-07]. Dostupný z WWW: <http://interval.cz/clanky/co-je-xml/>.

[35] JUJUJU. Co je to XML? [online]. 2002-06-16 [cit. 2008-04-10]. Dostupný z WWW: <http://www.pcsvet.cz/art/article.php?id=1830>.

[36] KOLÁŘ, David. Zápis správné syntaxe XML dokumentů [online]. 2001-01-22 [cit. 2008-04-14]. Dostupný z WWW: <http://www.builder.cz/art/html/xml_syntaxe.html>.

[37] JÍCHA, Radek. Využití XML (1.) [online]. 2003-01-22 [cit. 2008-04-14]. Dostupný z WWW: <http://www.pcsvet.cz/art/article.php?id=2971>.

[38] BRÁZDA, Jiří. XML praktické příklady. 1. vyd. Praha : Grada Publishing, a.s., 2003. 211 s. ISBN 80-247-0699-7.

[39] XML [online]. c1999-2008 [cit. 2008-04-10]. Dostupný z WWW: <http://www.symbio.cz/slovnik/xml.html>.

[40] Malware [online]. [2007] , Stránka byla naposledy editována 8. 5. 2008 [cit. 2008-05-08]. Dostupný z WWW: <http://cs.wikipedia.org/wiki/Malware>.

[41] BOTT, Ed, SIECHERT, Carl. Mistrovství v zabezpečení Microsoft Windows 2000 a XP. Praha : Computer Press, a.s., 2006. 696 s. ISBN 80-7226-878-3.

[42] Microsoft Corporation. Microsoft Windows XP Professional : Resource Kit. 1. vyd. Praha : Computer Press, a.s., 2002. 1468 s. ISBN 80-7226-608-X.

[43] Microsoft Corporation. Windows XP Professional : Jak na to [online]. c2008 [cit. 2008-05-08]. Dostupný z WWW: <http://www.microsoft.com/cze/windows/xp/pro/using/howto/default.mspx>.

[44] Microsoft Corporatin. Windows XP Professional : Centrum pro nápovědu a odbornou pomoc. 2002 [cit. 2008-05-08]..

[45] Opera's Kiosk Mode [online]. c2008 [cit. 2008-05-04]. Dostupný z WWW: <http://www.opera.com/support/mastering/kiosk/>.

[46] Opera's Command Line Options [online]. c2008 [cit. 2008-05-04]. Dostupný z WWW: <http://www.opera.com/docs/switches/>.

[47] Adobe Flex [online]. c1999-2008 [cit. 2008-05-10]. Dostupný z WWW: <http://www.symbio.cz/slovnik/adobe-flex.html>.

[48] BERNARD, Borek. Rich Internet Applications v roce 2008 [online]. 2008-04-25 [cit. 2008-05-11]. Dostupný z WWW: <http://interval.cz/clanky/rich-internet-applications-v-roce-2008/>.

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

ADC	Analog to Digital Converter
AIR	Adobe Integrated Runtime
AJAX	Asynchronous JavaScript and XML
AVM	ActionScript Virtual Machine
CD-ROM	Compact Disc – Read Only Memory
CLR	Common Language Runtime
CRT	Cathode Ray Tube
CSS	Cascading Style Sheets
DHTML	Dynamic HTML
DLR	Dynamic Language Runtime
DOM	Document Object Model
DTD	Document Type Definition
DVD	Digital Versatile Disc
ECMA	European Computer Manufacturers Association
FTP	File Transfer Protocol
GPL	General Public License
HD	High Definition
HTML	HyperText Markup Language
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol over Secure Socket Layer
IDE	Integrated Development Environment
IE	Internet Explorer
IP	International Protection
IR-LED	Infrared - Light Emitting Diode

ITO	Indium Tin Oxide
Java ME	Java Micro Edition
Java RPC	Java Remote Procedure Call
Java SE	Java Standard Edition
JSON	JavaScript Object Notation
LCD	Liquid Crystal Display
LINQ	Language Integrated Query
PC	Personal Computer
PCI	Peripheral Component Interconnect
PDA	Personal Digital Assistant
PDF	Portable Document Format
PHP	PHP: Hypertext Preprocessor
POX	Plain Old XML
RIA	Rich Internet Application
REST	Representational State Transfer
RSS	Really Simple Syndication
SAW	Surface Acoustic Wave
SDK	Software Development Kit
SGML	Standard Generalized Markup Language
SOAP	Simple Object Access Protocol
SP	Service Pack
SVG	Scalable Vector Graphics
URL	Uniform Resource Locator
USB	Universal Serial Bus
W3C	World Wide Web Consorcium

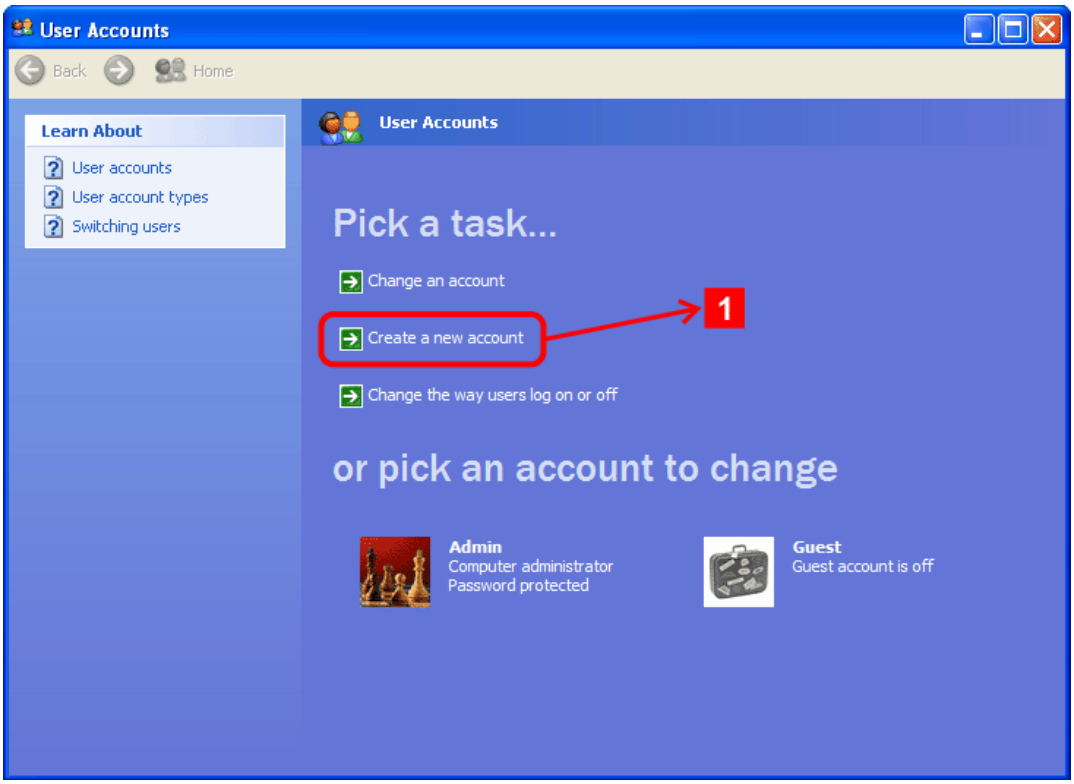
WMA	Windows Media Audio
WPF/e	Windows Presentation Foundation Everywhere
XAML	Extensible Application Markup Language
XHTML	Extensible HyperText Markup Language
XML	Extensible Markup Language
XML-RPC	XML- Remote Procedure Call
XSD	XML Schema Definition
XSL	Extensible Stylesheet Language
XSLT	Extensible Stylesheet Language Transformations

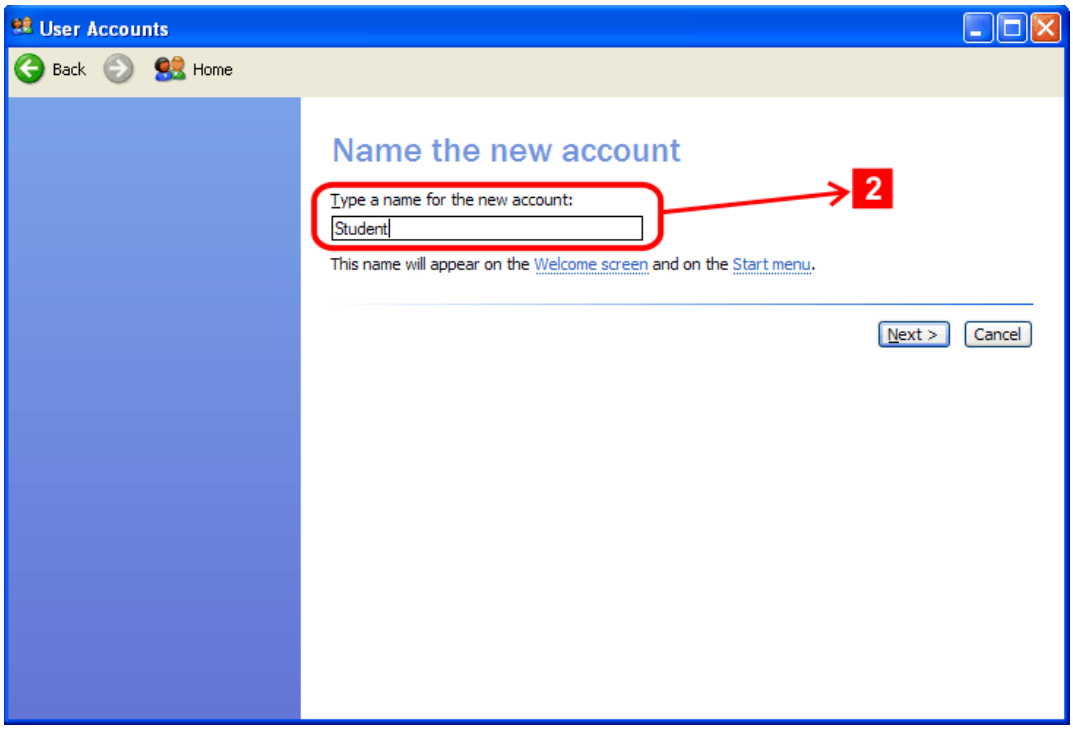
SEZNAM OBRÁZKŮ

SEZNAM TABULEK

SEZNAM PŘÍLOH

PŘÍLOHA P I: VYTVOŘENÍ UŽIVATELSKÉHO ÚČTU



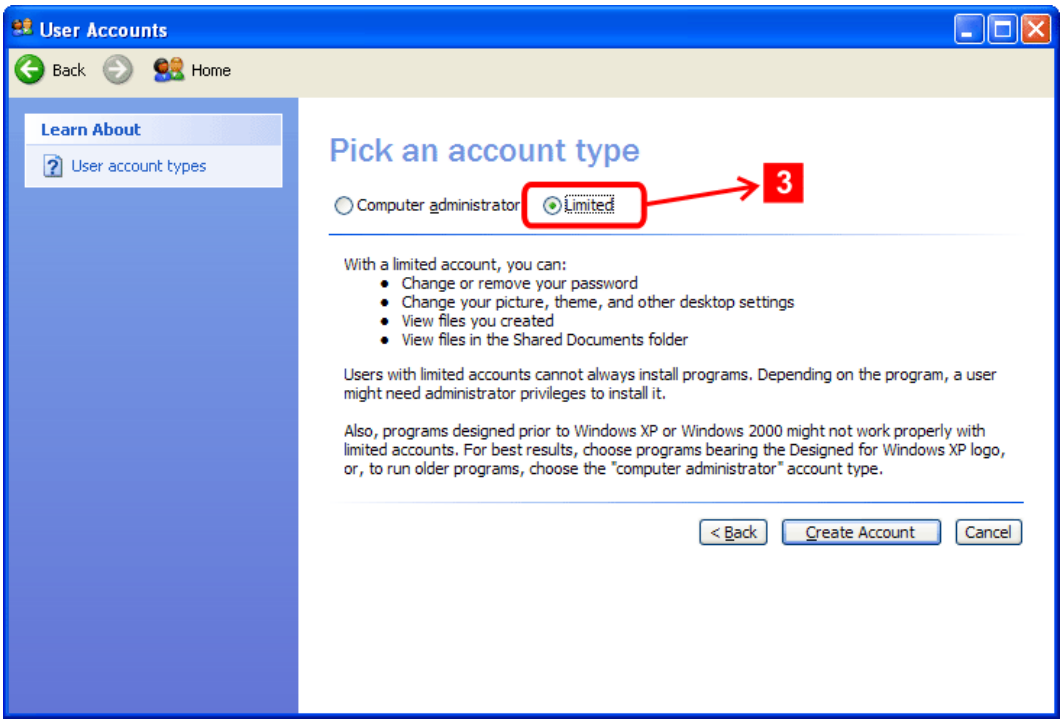


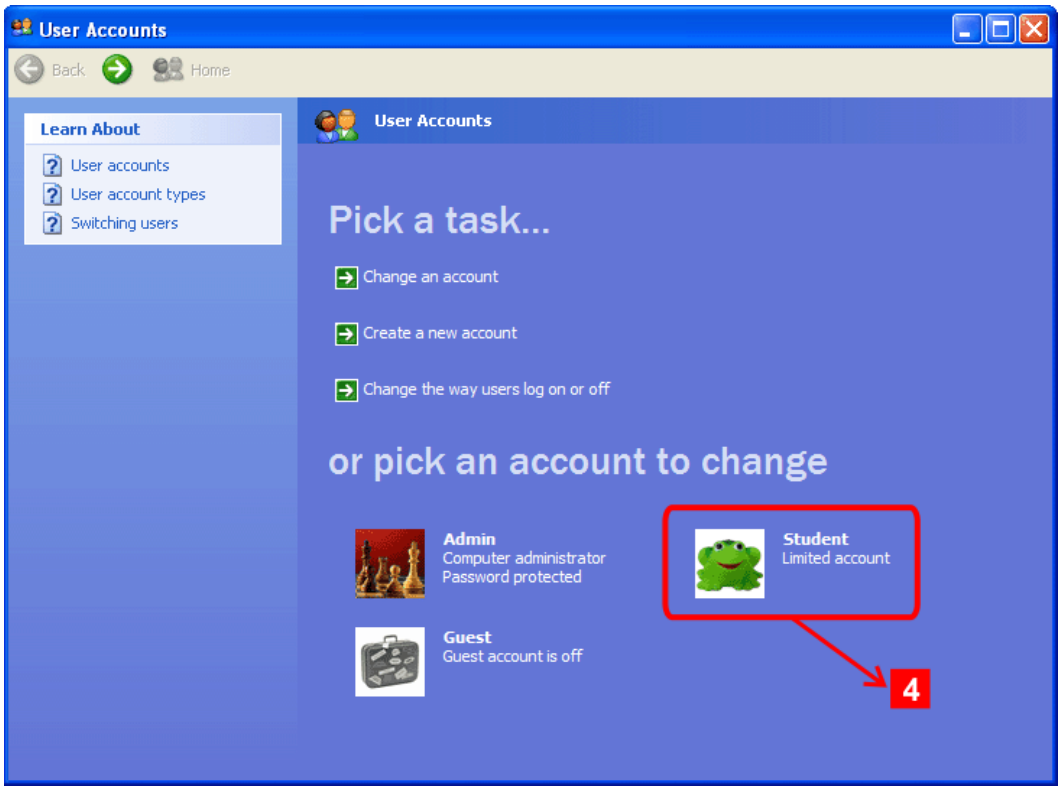
Vysvětlivky:

1 – Vytvoření nového uživatelského účtu.

2 – Název nového uživatelského účtu.

PŘÍLOHA P II: VÝBĚR TYPU ÚČTU A PŘEHLED ÚČTŮ



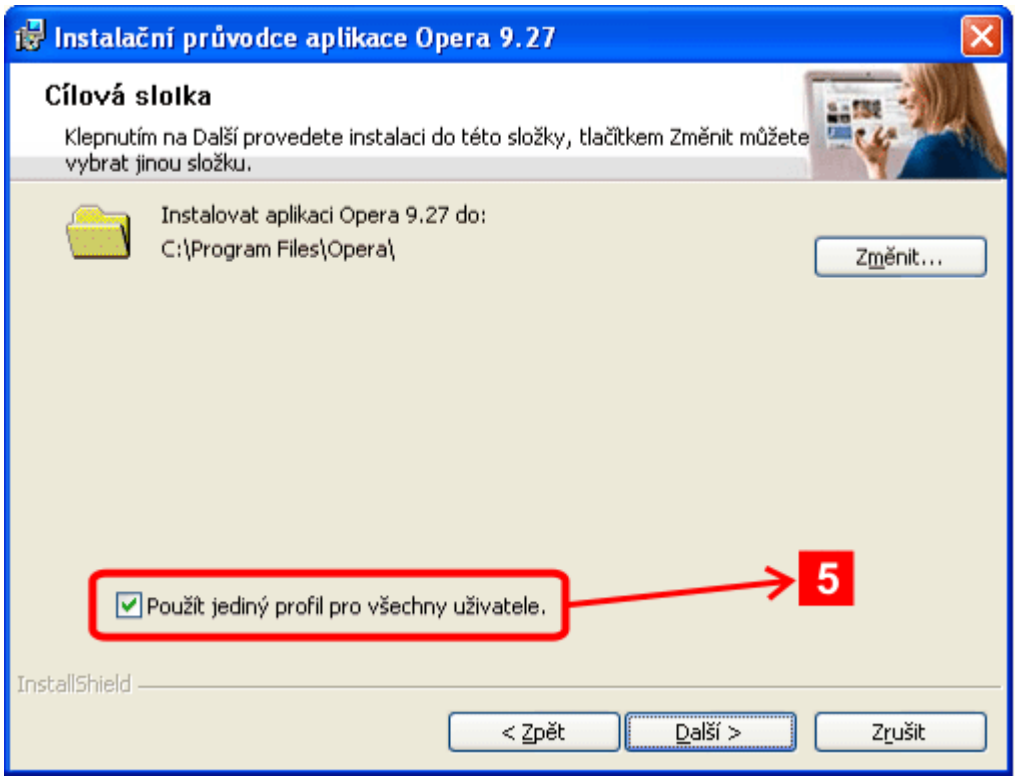


Vysvětlivky:

3 – Volba typu účtu s omezenými právy.

4 – Přehled všech účtů na počítači, včetně nově vytvořeného účtu "Student".

PŘÍLOHA P III: INSTALACE PROHLÍŽEČE OPERA



Vysvětlivky:

5 – Zaškrtnutím volby "Použít jediný profil pro všechny uživatele" může správce po úspěšné instalaci prohlížeče provádět úpravy vzhledu a funkčnosti pouze jednou. Provedené změny se projeví i v účtech uživatelů.

PŘÍLOHA P IV: WINDOWS STEADYSTATE – HLAVNÍ NABÍDKA

Vysvětlivky:

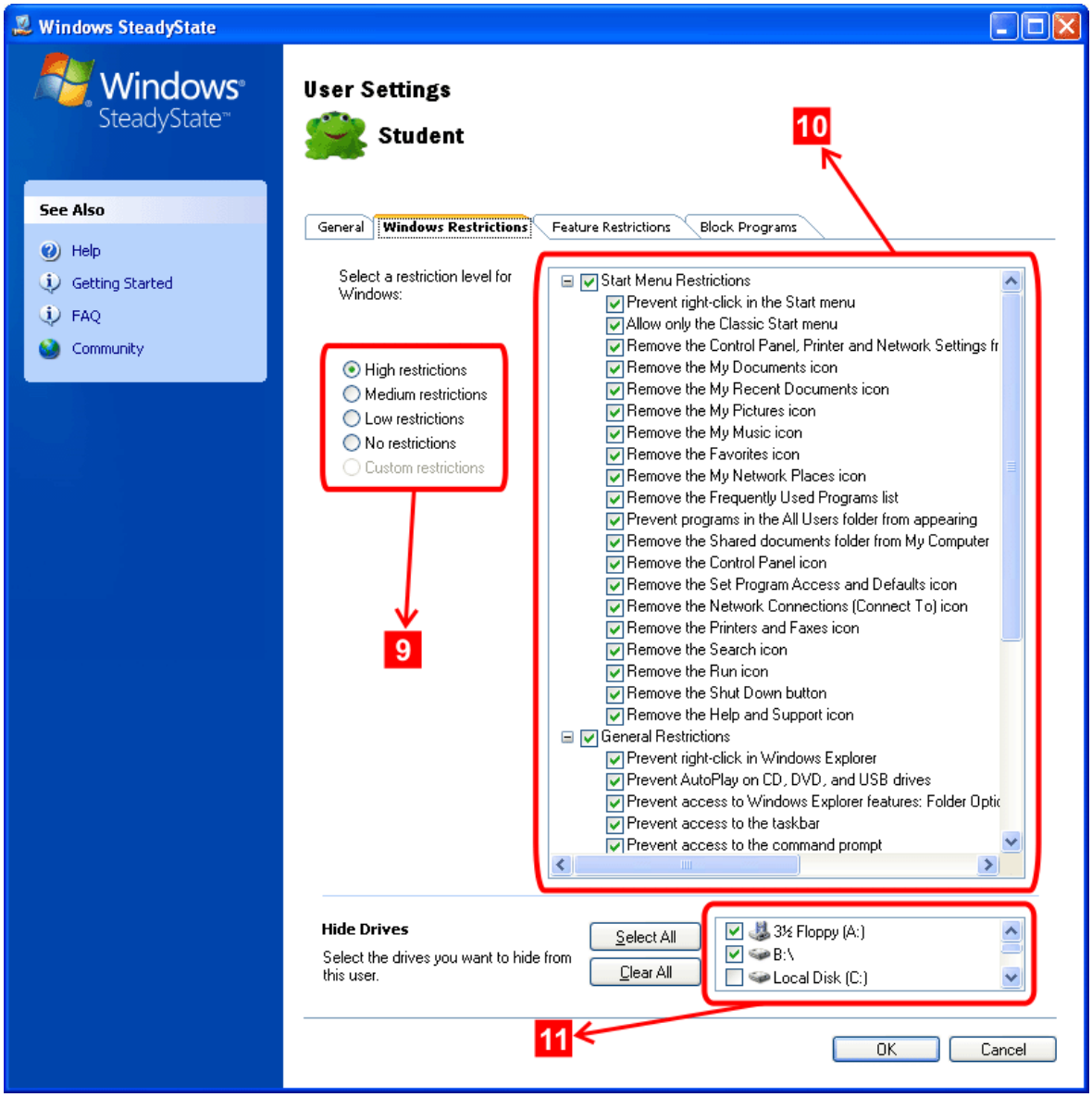
Úvodní okno programu Windows SteadyState.

6 – Odkaz na nastavení zabezpečení účtu uživatele "Student" omezením některých funkcí.

7 – Odkaz na volby omezení účtů všech uživatelů ze skupiny "Users".

8 – Možnosti ochrany dat uložených na pevném disku.

PŘÍLOHA P V: WINDOWS STEADYSTATE – OMEZENÍ UŽIVATELSKÉHO ÚČTU



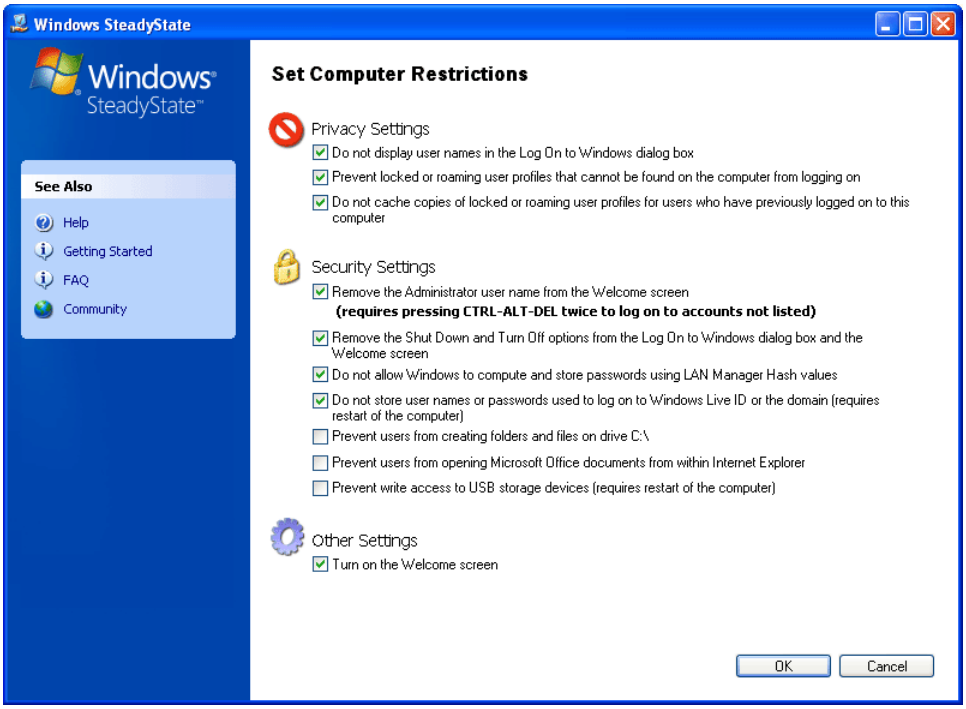
Vysvětlivky:

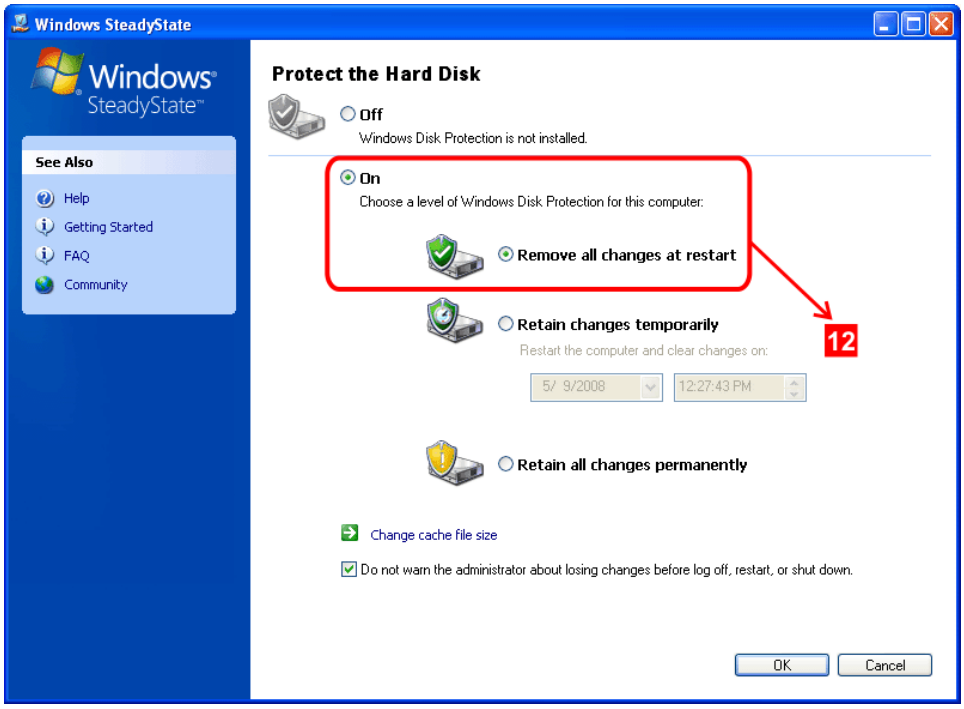
9 – Volba úrovně omezení, resp. zabezpečení.

10 – Seznam možných omezení.

11 – Zákaz zobrazení určitých pevných disků a mechanik pružných disků.

PŘÍLOHA P VI: WINDOWS STEADYSTATE – OMEZENÍ ÚČTŮ A OCHRANA DISKU



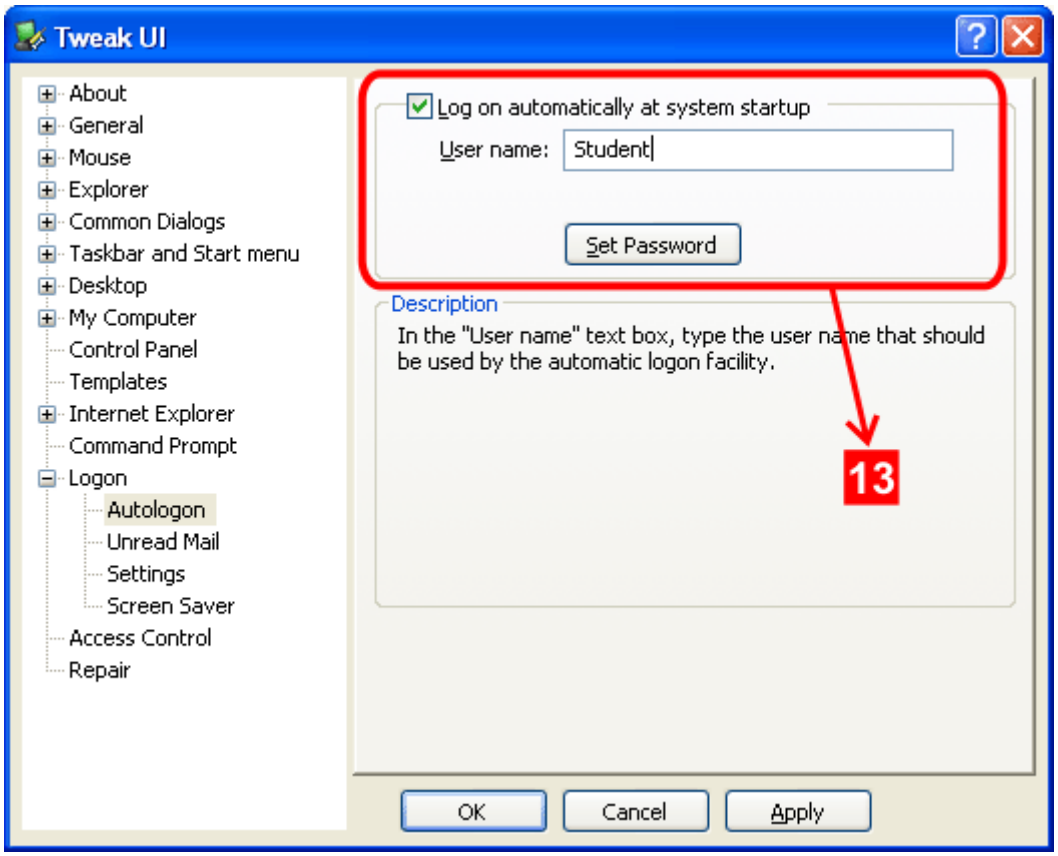


Vysvětlivky:

Volby omezení všech uživatelských účtů ve skupině "Users".

12 – Zapnutí ochrany pevného disku (během práce uživatele na počítači se data ukládají do cache paměti a po odhlášení se účet vrátí do původního stavu).

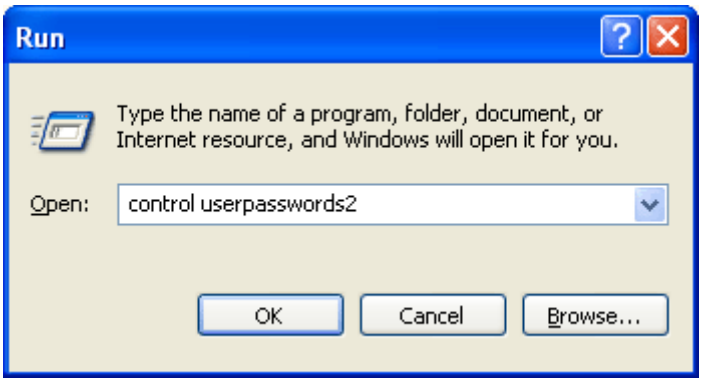
PŘÍLOHA P VII: TWEAK UI - AUTOLOGON

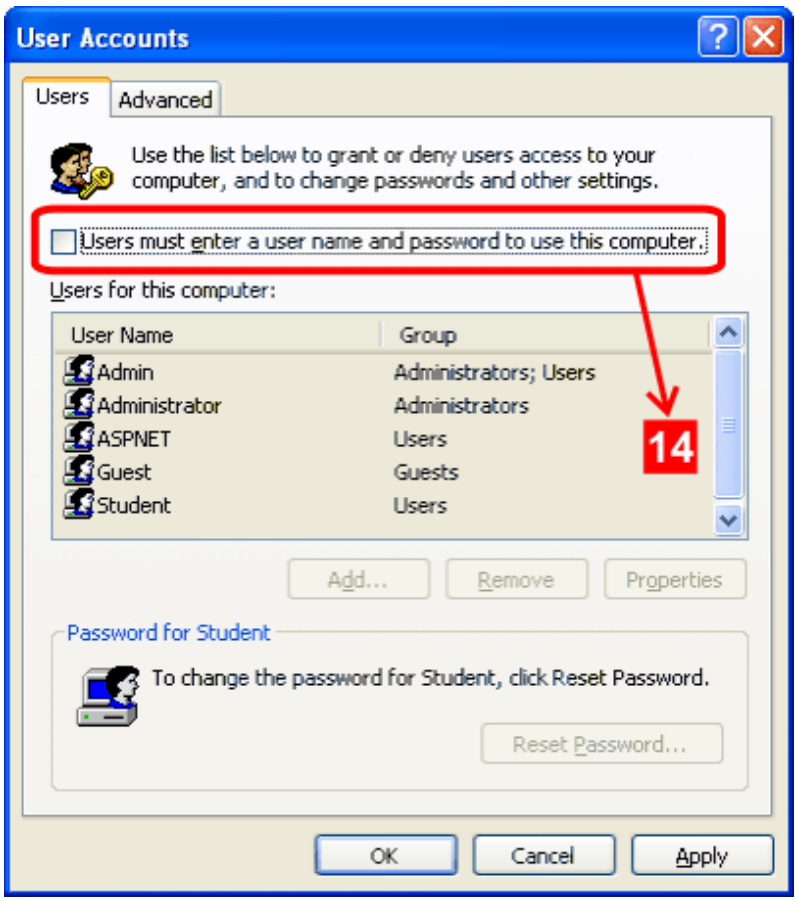


Vysvětlivky:

13 – Nastavení automatického přihlášení uživatele "Student".

PŘÍLOHA P VIII: PŘÍKAZ "CONTROL USERPASSWORDS2"





Vysvětlivky:

Spuštění příkazu pomocí *Start -> Spustit… -> control userpasswords2*

14 – Odškrtnutím vyznačeného pole se deaktivuje přihlašovací obrazovka se jmény uživatelů.

Další možností programu spuštěného tímto příkazem je rozšířená správa uživatelských účtů.