

Univerzita Komenského v Bratislave  
Fakulta matematiky, fyziky a informatiky  
Katedra informatiky



## **Knižnica (framework) pre kreslenie grafov**

Diplomová práca

Bc. Tomáš DRIMAL

Študijný odbor: 9.2.1 Informatika

Vedúci práce: RNDr. Richard Ostertág

BRATISLAVA 2009

**Prehlásenie:**

Týmto prehlasujem, že som diplomovú prácu vypracoval samostatne a uvádzam všetku použitú literatúru.

.....

Ďakujem svojmu diplomovému vedúcemu RNDr. Richardovi Ostertágovi za cenné rady a pomoc pri písaní diplomovej práce. Taktiež chcem poďakovať svojim blízkym za ich podporu počas písania tejto práce.

# Abstrakt

Cieľom práce bolo urobiť prehľad technológií, pomocou ktorých by sa dal implementovať interaktívny systém na zobrazovanie grafov. Cieľom bolo taktiež zanalyzovať dostupné riešenia, určiť ich nedostatky, nájsť prednosti a zaujímavé myšlienky, ktoré by sa dali následne uplatniť pri návrhu vlastného interaktívneho systému. Cieľom bolo navrhnutý systém aj implementovať a vytvoriť web so systémom, dokumentáciou a ukázkami na stiahnutie.

**Kľúčové slová:** vykresľovanie grafov, Silverlight, C#, interaktívny systém

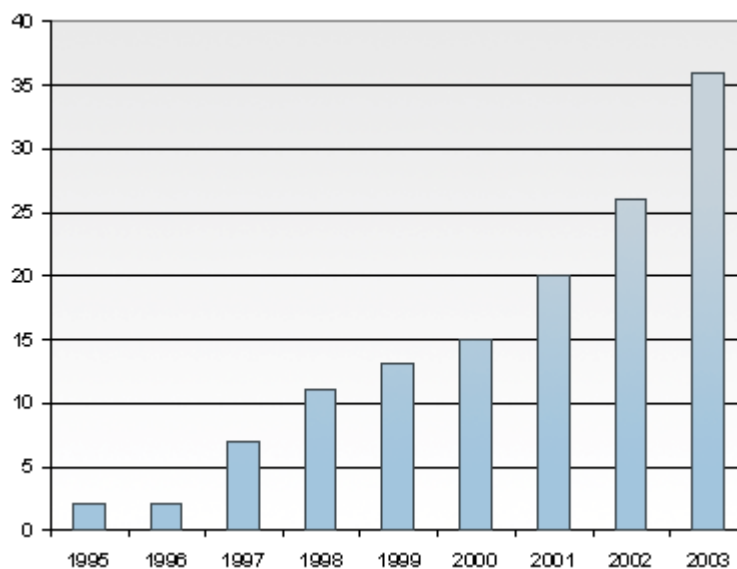
# Obsah

<b>1. Úvod</b>	<b>6</b>
1.1. Motivácia	6
1.2. Cieľ práce	7
1.3. Členenie práce	7
<b>2. Možnosti implementácie</b>	<b>8</b>
2.1. Bitmapa	8
2.2. HTML tag Canvas	8
2.3. SVG a VML	9
2.4. Adobe Flash	11
2.5. Adobe Flex	12
2.6. Microsoft Silverlight	13
2.6.1. Charakteristika	13
2.6.2. Multiplatformovosť a perspektíva	15
2.6.3. Bezpečnosť	17
2.6.4. XAML	17
<b>3. Dostupné riešenia</b>	<b>22</b>
3.1. GigaSoft ProEssentials Charts	22
3.2. Telerik RadChart for ASP.NET AJAX	23
3.3. Chart FX	25
3.4. Fusion Charts	26
3.5. Dundas Chart	27
3.6. Visifire	28
3.7. Rozšírený prehľad riešení	29
3.7.1. Flash	29
3.7.2. PHP a ASP knižnice	32
3.7.3. JavaScript	34
<b>4. Návrh a realizácia riešenia</b>	<b>36</b>
4.1. Špecifikácia	37
4.1.1. Dátový zdroj	37
4.1.2. Vykresľovanie grafov	38
4.1.3. Prepojenia grafov	38
4.1.4. Zoskupovanie dát v grafoch	38
4.1.5. Zorad'ovanie	39
4.1.6. Funkcia ZOOM a Full Screen	39
4.1.7. Save a Load	39
4.1.8. ScaleBreak	40
4.1.9. Implementácia do stránok	40

4.2. Návrh užívateľského prostredia	41
4.2.1. Menu	42
4.2.2. Vykresľovanie grafov	43
4.2.3. Zoskupovanie dát v grafoch	44
4.2.4. Zorad'ovanie	45
4.2.5. Funkcia ZOOM a Full Screen	45
4.2.6. ScaleBreak	46
4.3. Implementácia	48
4.3.1. Fyzická štruktúra	48
4.3.2. Popis súborov a tried	48
4.3.3. Dátový zdroj	50
4.3.3.1. Tag <settings>	51
4.3.3.2. Tag <datasection>	54
4.3.3.3. LINQ	54
4.3.4. Vykresľovanie grafov	56
4.3.5. Prepojenia grafov	57
4.3.6. Zoskupovanie dát v grafoch	57
4.3.7. Zorad'ovanie	57
4.3.8. Funkcia ZOOM a Full Screen	58
4.3.9. Save a Load	59
4.3.10. ScaleBreak	60
4.3.11. Implementácia do stránok	61
<b>5. Záver</b>	<b>63</b>
<b>6. Zoznam použitej literatúry</b>	<b>64</b>

# 1. Úvod

Pod pojmom graf v tejto diplomovej práci rozumieme znázornenie priebehu diskkrétnej funkcie na konečnom intervale. Ako napríklad stĺpcový graf na obr.1.



Obr. 1: Ukážka stĺpcového grafu

## 1.1 Motivácia

Pôvodným cieľom bolo navrhnuť systém na vykresľovanie grafov, ktorý by bol použiteľný ako na desktopové aplikácie, tak aj na internetové. Avšak vzhľadom na rozsah problematiky sme sa rozhodli zamerať na internetové/intranetové stránky (ďalej iba stránky).

Väčšina dostupných riešení, ktoré majú za účel komplexne vykresľovať rôzne druhy grafov na stránkach, je dostupných iba vo forme komponentov (napríklad do Visual Studio), prípadne ako knižnice, čo znemožňuje ich použitie pre širšiu verejnosť, keďže je potrebná lepšia znalosť niektorého z programovacích jazykov.

Po analýze dostupných riešení sme dospeli k záveru, že momentálne nie je systém, ktorý by umožňoval na webe vykresľovať grafy a spĺňal nasledujúce body:

- Umožňoval vzájomnú interakciu viacerých grafov súčasne.
- Bol jednoducho použiteľný (aby implementáciu do svojich stránok zvládol aj menej skúsený tvorca internetových stránok).
- Bol zadarmo aj pre komerčné účely.

## 1.2 Cieľ práce

- Urobiť prehľad dostupných riešení, techník a technológií, ktoré umožňujú vykresľovanie grafov na stránkach.
- Analyzovať dostupné riešenia, určiť ich nedostatky, nájsť prednosti a zaujímavé myšlienky, ktoré by sa dali následne uplatniť pri návrhu nášho systému.
- Pri analýze sa zamerať hlavne na možnosti interakcie s užívateľom a jednoduchosť implementácie do stránok.
- Navrhnuť a implementovať systém na kreslenie grafov, ktorý by mal mať zaujímavé a zároveň prehľadné užívateľské rozhranie a ktorý by umožňoval:
  - Interakciu s užívateľom.
  - Vykresľovanie súčasne viacerých grafov a ich vzájomne ovplyvňovanie.
  - Veľmi jednoduché zakomponovanie do stránok.
  - Systém vydať pod free licenciou.
- Vytvoriť web so systémom, dokumentáciou a ukážkami.

## 1.3 Členenie práce

Práca je členená na tri hlavné kapitoly. Prvá popisuje možnosti/techniky, pomocou ktorých je možné vykresľovať grafy na stránkach. Sú v nej ukázané ich výhody, nevýhody. Táto kapitola nemá za cieľ do detailov predstaviť každú z techník, ale priblížiť ich fungovanie, aby sa dala urobiť predstava o ich „sile“. Podrobnejšie je popísaná technológia Silverlight 2, pomocou ktorej sme sa rozhodli implementovaný náš systém.

V druhej kapitole sú predstavené niektoré dostupné komerčné aj nekomerčné riešenia, ktoré vedia vykresľovať grafy na stránkach, pričom sa zameriava na ich prednosti, nedostatky. Vyzdvihujú sa tam niektoré zaujímavé koncepty, ktoré boli inšpiráciou pri samotnej implementácii systému.

Posledná kapitola je venovaná návrhu a realizácii samotného systému. Je ďalej rozdelená na podkapitoly, venované špecifikácií, návrhu užívateľského prostredia a nakoniec aj samotnej implementácii.



## 2. Možnosti implementácie

### 2.1. Bitmapa

Ako prvú možnosť použiteľnú na generovania grafov sme vybrali zobrazenie grafov ako obyčajných obrázkov. Toto riešenie je hojne používané rôznymi aplikáciami, ktoré sú napísané v C#, Jave, PHP, ASP.NET,.... Taktiež sú knižnice, ktoré obsahujú metódy na kreslenie takýchto grafov.

Samotná implementácia takéhoto obrázku do svojich stránok je veľmi jednoduchá. Spočíva napríklad iba v umiestnení HTML tagu

```

```

do svojich stránok. Kompatibilita s prehliadačmi je 100%, keďže ide iba o zobrazenie obrázku v na to určenom HTML tagu. Najväčší nedostatok tohto riešenia je v slabej a zložitej implementácii interakcie s užívateľom. Obrázok by musel byť klikacia mapa a po každom kliknutí by bol poslaný dotaz na server a ten následne vygeneroval nový obrázok. Server by bol zbytočne často vyťažovaný takýmito dotazmi.

### 2.2. HTML tag Canvas

Ďalšou možnosťou ako vykresľovať grafy na stránkach je možnosť využiť HTML prvok `<canvas></canvas>`, do ktorého je umožnené kresliť pomocou JavaScriptu.

Tento prvok bol predstavený firmou Apple pre Dashboard v Mac OS X a následne ho implementovali aj do svojho internetového prehliadača Safari. Internetové prehliadače, ktoré sú založené na jadre Gecko 1.8 a novšom už podporujú tento tag. Napríklad Firefox 1.5, ako aj Opera od verzie 9 a prehliadač Safari. Tento prvok však nie je podporovaný všetkými prehliadačmi. Momentálne najpoužívanejší Internet Explorer má s týmto tagom problémy. Vie ho však emulovať pomocou ActiveX komponentu DirectAnimation, alebo cez VML. Tag canvas sa stal aj súčasťou HTML 5.

```

1 <html>
2 <head>
3   <script type="application/x-javascript">
4     function draw() {
5       var canvas = document.getElementById("canvas");
6       var ctx = canvas.getContext("2d");
7
8       ctx.beginPath();
9       ctx.arc(75,75,50,0,Math.PI*2,true);
10      ctx.moveTo(110,75);
11      ctx.arc(75,75,35,0,Math.PI,false);
12      ctx.moveTo(65,65);
13      ctx.arc(60,65,5,0,Math.PI*2,true);
14      ctx.moveTo(95,65);
15      ctx.arc(90,65,5,0,Math.PI*2,true);
16      ctx.stroke();
17    }
18  </script>
19 </head>
20 <body onload="draw()">
21   <canvas id="canvas" width="300" height="300"></canvas>
22 </body>
23 </html>
24

```



Obr. 2 Ukážka použitia tagu Canvas a výsledok

Na prvý pohľad je táto možnosť generovania obrázkov lepšia ako Bitmapa, lebo pomocou JavaScriptu už vieme zabezpečiť jednoduchšiu interaktivitu. Podpora tohto tagu v internetových prehliadačoch je však kvôli veľkému podielu Internet Exploreru, ktorý ho nepodporuje, stále nedostatočná. Možnosti interaktivity sú tiež slabé. Preto túto možnosť nepovažujeme za vhodnú.

## 2.3. SVG a VML

VML (*Vector Markup Language*) aj SVG (*Scalable Vector Graphic*) sú značkové jazyky a formát súborov, ktoré pomocou XML popisujú dvojrozmernú grafiku. VML aj SVG sú si veľmi podobné, pričom VML bolo v roku 1998 predstavené Microsoftom a má natívnu podporu v prehliadači Internet Explorer. SVG je otvorený formát, ktorý je nástupcom VML a ktorý W3C odporúča už od roku 2001 ako otvorený štandard pre vektorovú grafiku. Ďalej sa preto budeme zaoberať už iba SVG jazykom. Internet Explorer nepodporuje natívne SVG. Na zobrazenie SVG súborov je potrebné nainštalovať plugin Adobe SVG Viewer od spoločnosti Adobe, ktorá 1.1.2009 ukončila podporu pre tento plugin.

Nižšie môžete vidieť ukážku kódu jednoduchého obrázku v SVG, ktorý vykreslí červenú hviezdu.

```

<?xml version="1.0"?>
<svg xmlns="http://www.w3.org/2000/svg">
  <title>Hviezda</title>
  <polygon style="fill:red; stroke:black; stroke-width:7"
    points="350, 75 379,161 469,161 397,215
           423,301 350,250 277,301 303,215
           231,161 321,161" />
</svg> |

```



Obr. 3: Ukážka použitia SVG a výsledok

SVG má tri základné tvary objektov a to vektorové tvary, rastrové obrazy a textové objekty, ktoré môžu byť rôzne zoskupované, formátované pomocou atribútov, alebo CSS. Objekty sú vykresľované v poradí, v akom sú zapísané v zdrojovom kóde a môže sa im definovať priehľadnosť. Text rovnako ako v HTML zostáva stále textom s možnosťou fulltextového vyhľadávania, alebo kopírovania. Samozrejmosťou je podpora kódovania UNICODE.

Zaujímavosťou je, že cez štandardné rozhranie (DOM) sa dá pristupovať k ľubovoľnému objektu, meniť ho, prípadne vytvárať úplne nové objekty. Taktiež fungujú štandardné mechanizmy udalostí (events).

Z doposiaľ menovaných riešení je SVG prvý kandidát nad ktorým by sa dalo uvažovať ako o nástroji na vykresľovanie grafov na webe. Ponúka robustné možnosti práce s grafikou a taktiež jeho súčasná podpora v prehliadačoch je dobrá, všetky súčasné prehliadače vedú zobrazovať SVG, výnimkou je (typicky) Internet Explorer, ktorý doposiaľ natívne podporuje VML a SVG je podporované iba s doplnkovým pluginom.

Na druhej strane SVG je primárne určené na popis vektorovej grafiky a nie na tvorbu užívateľského prostredia. Samotné grafy, keďže sa jedná o vektorovú grafiku, by boli veľmi pekné, ostré, neboli by rozmazané. Taktiež ukončená podpora ďalšieho vývoja pluginu pre Internet Explorer hovorí v neprospech tejto technológie.

## 2.4. Adobe Flash

Dostávame sa k prvému silnému kandidátovi, pomocou, ktorého by sa už dal vytvoriť dostatočne robustný a interaktívny systém na zobrazovanie grafov a to k Adobe Flash, s ktorým sa už určite každý na internete stretol.

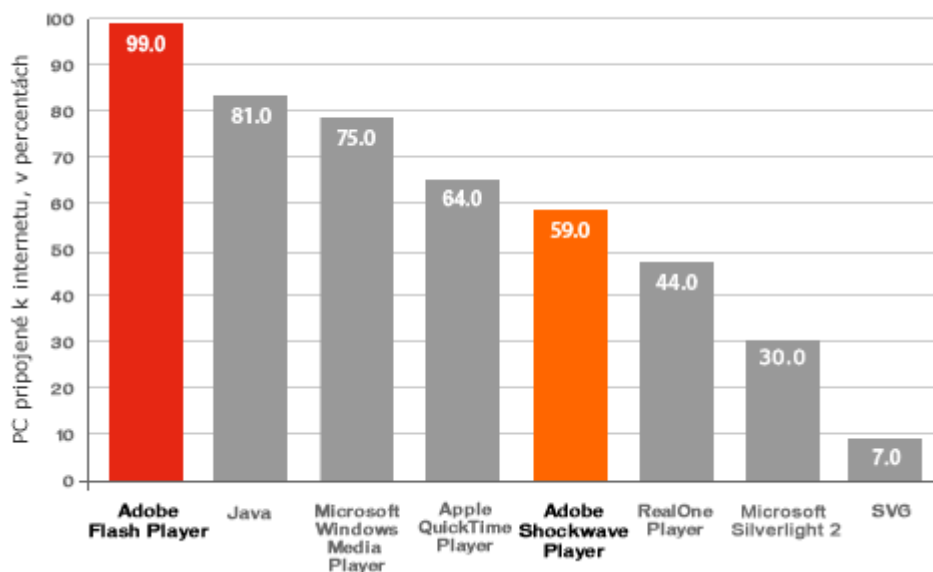
Flash je technológia, pomocou ktorej sa dajú vytvárať animácie a vysoko interaktívne audiovizuálne prezentácie. Bola predstavená v roku 1996 a odvtedy získala obrovskú popularitu pri tvorbe animovaného a interaktívneho webového obsahu. Veľkou výhodou je, že sa tvorca aplikácie vie vždy spoľahnúť na to, že jeho výtvor sa bude všade zobrazovať rovnako. Toto ma na starosti plugin (Adobe Flash Player), ktorý sa inštaluje do prehliadača.

Nevýhodou je nedostupnosť free programov, ktoré by umožňovali vytvárať Flash animácie. Existuje niekoľko free programov, ktoré sú však na veľmi nízkej úrovni. Preto treba pri tvorbe siahnuť po programe Adobe Flash (momentálne vo verzii CS4), pomocou ktorého sa dajú naplno využiť všetky možnosti tejto technológie. Toto vývojové prostredie je však komerčné. Taktiež Flash technológia je veľmi špecializovaná a prácu s ňou sa treba kompletne od základu naučiť.

Tvorba projektu by sa dala zhrnúť do fázy designovania a následného naprogramovania funkcionality. Na programovanie sa využíva ActionScript (momentálne verzia 3.0, podporovaná Adobe Flash Playerom 9), čo je objektovo orientovaný jazyk pre aplikácie, ktoré sú vyvíjane pomocou Adobe Flash. ActionScript vychádza zo štandardizovanej verzie jazyka JavaScript nazvanej ECMAScript. ActionScript nie je vždy potrebný, ale pri používaní interaktivity ako sú klikania myši a rôzne udalosti je nutný.

V prospech tejto technológie hovorí široká možnosť interaktivity, ako aj rozšírenosť vo svete. Ako vidieť z obr.4 až 99% PC, ktoré sú pripojené k internetu majú podľa štatistiky nainštalovaný Adobe Flash PLayer.

Na druhej strane v neprospech hovorí neefektívnosť tvorby. Keďže ide o špecifickú komerčnú technológiu, bolo by nutné sa naučiť kompletne logiku vývojového prostredia a ActionScript.



Obr. 4: Rozšírenosť Adobe Flash Playera a iných technológií (December 2008)

## 2.5. Adobe Flex

Dostávame sa k jednej z najnovších technológií ktorá má názov Adobe Flex. Zjednodušene sa dá povedať, že Flex je vývojová technológia pre RIA (Rich Internet Application). Je to technológia pre vývoj aplikácií (nie bannerov, animácií, alebo prezentačných stránok). Je to sada komponentov a tried spolu s kompilátormi a command-line nástrojmi. Toto všetko ako celok sa nazýva Flex SDK. Je to iba vývojové prostredie, pričom v prehliadači sa na beh Flex aplikácií používa Adobe Flash Player a na desktope slúži k spusteniu Adobe AIR.

Môže sa zdať že Flex je veľmi podobný Flashu, vďaka tomu že oba bežia pod Flash Playerom, čo je aj jedna z najväčších výhod Flexu, ale opak je pravdou. Flex nemá s Flashom okrem Flash Playera spoločné nič.

Flash slúži na tvorbu animácií, bannerov a medzi jeho základný koncept patrí časová os. Na druhej strane Flex nič také ako časovú os nepozná, ide o programátorské rozhranie. Flash aplikácie sa dajú vyvíjať iba vo Flashi, naproti tomu Flex môžeme vyvíjať aj v jednoduchom textovom editore a kompilovať z príkazového riadka (komfortnejšie je však využívať vývojové prostredie Flex Builder). Ďalej tiež zdrojové kódy Flashu sú v binárnej podobe a k ich editácii je nutný Flash IDE, naproti tomu zdrojové kódy Flexu sú obyčajné textové súbory.

Adobe Flex sa prepracoval na hlavného kandidáta pre tvorbu systému na vykresľovanie grafov.

## 2.6. Microsoft Silverlight

Microsoft Silverlight je veľmi podobná technológii Adobe Flex. Obe taktiež prišli na trh v približne rovnakom čase a ponúkajú podobnú funkcionálnosť, aj keď každá trochu iným spôsobom. Táto kapitola čerpá informácie z [9, 4, 6].

Obe technológie sú akousi reakciou na momentálny trend na internete, ktorý sa označuje „Web 2.0“. Je to veľmi často používaný pojem. Dá sa povedať, že ide o dynamické, interaktívne webové aplikácie, ktoré umožňujú svoje prispôbenie užívateľmi. Presnú definíciu je však veľmi ťažko sformulovať, lebo nie je. Web 2.0 by malo označovať niečo ako „popadnutie druhého dychu“ pre internetové spoločnosti. Nejde tu o nejaký technologický termín, ktorý by označoval nejakú zmenu štandardov. Ide o zmenu prístupu k ľuďom. Zaujímavá je definícia blogera Rossa Mayfielda: „*Web 1.0 was commerce. Web 2.0 is people.*“

Postupným vývojom internetových/intranetových aplikácií sa docielil stav, kedy majú používatelia stále väčšie nároky na používateľské rozhranie, ktoré by sa malo čo najviac priblížiť možnostiam klasických „tlstých“ aplikácií. Aplikácie, ktoré sú však postavené na klasických webových technológiách narádzajú pri tvorbe interaktívnych užívateľských rozhraní na mantinely použitých technológií. Hlavným problémom je potreba pri zmene stavu znova načítavať obsah, ako aj problém so zapamätávaním si stavových informácií pri používaní bezstavového HTML protokolu. Jedno z riešení je rozšíriť prehliadač o interaktívnu prezentačnú vrstvu, pre ktorú sa ujalo označenie Rich Presentation Tier. A preto prišiel Microsoft so Silverlightom, ktorý rozširuje prezentačnú úroveň prehliadača o nové možnosti s využitím vektorovej grafiky.

### 2.6.1. Charakteristika

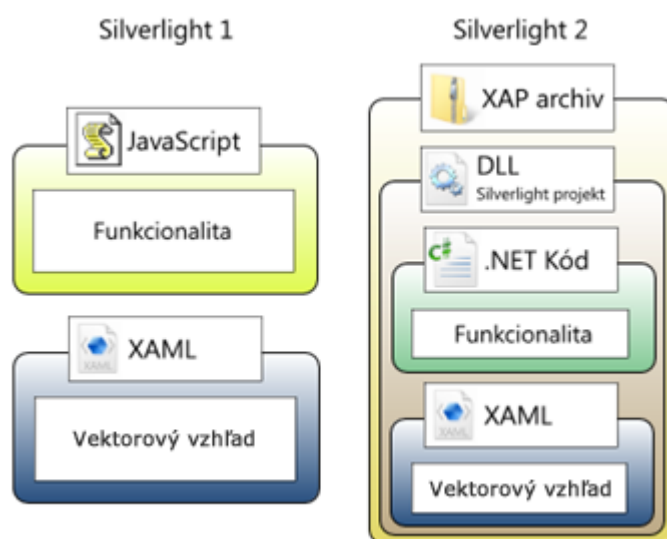
V jednej vete je Silverlight technológia, ktorá je určená k bezpečnému prehrávaniu interaktívnych prezentácií v prehliadači klienta na rôznych platformách (čím sa stáva konkurentov Flashu, Flexu). Je veľmi podobný technológii WPF (Windows Presentation Foundation). V skutočnosti je to jeho odľahčená verzia, ktorá je určená pre web.

WPF je nový grafický framework, ktorý sa používa pri vývoji Windows aplikácií. Ponúka mnoho nových, grafických možností, ktoré boli pri WinForms nemysliteľné. Je súčasťou .NET Frameworku 3.0 a táto platforma je štandardne predinštalovaná v operačnom systéme Windows Vista.

Prvá finálna verzia Microsoft Silverlight 1.0 bola vydaná 5.9.2007. V rovnakom čase bola tiež predstavená alfa verzia Silverlight 1.1, ktorá sa neskôr pri vydaní, v septembri 2008, zmenila na Silverlight 2. Budeme sa zaoberať hlavne popisom verzie Silverlight 2, ale najprv ukážem niektoré

zásadne rozdiely medzi týmito verziami. Silverlight 1 je niečo ako predskokana k verzii 2, lebo až tá využíva plnohodnotne svoj potenciál.

Hlavným rozdielom medzi Silverlight 1.0 a Silverlight 2 je možnosť písania menežovateľného kódu v niektorom z .NET jazykov. V Silverlight 1.0 to bolo možné iba pomocou JavaScriptu, čo mnohých odrádzalo od jeho použitia. Taktiež pribudli oproti prvej verzii niektoré nové ovládacie prvky, z ktorých niektoré považujeme v súčasnosti za samozrejmosť ako napríklad Button, ListBox, GridView, TextBox,....

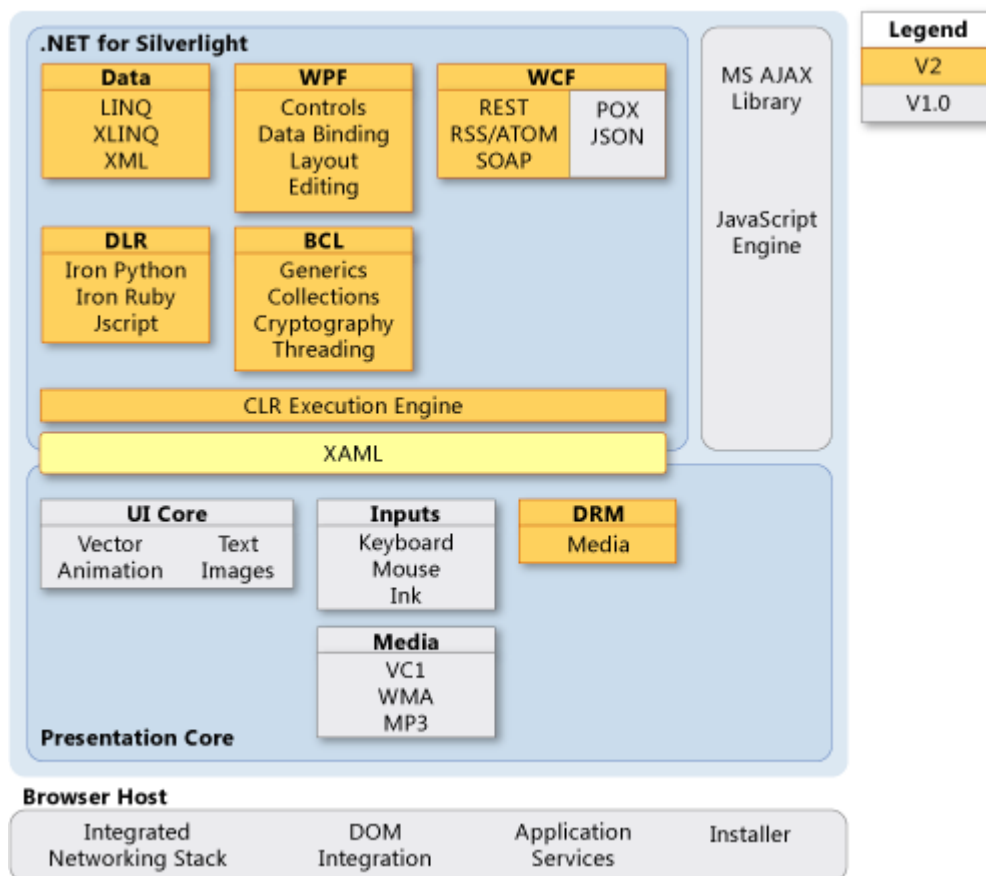


Obr. 5: Rozdiel medzi Silverlight 1.0 a Silverlight 2

Niektoré hlavné charakteristiky Silverlight 2:

- Rozšírená podpora .NET Framework Language: Silverlight 2 podporuje celý rad programovacích jazykov okrem iného aj: Visual Basic, C#, JavaScript, IronPython a IronRuby. Je to veľká výhoda, keď zoberieme do úvahy, že vývojári môžu svoje súčasné skúsenosti a schopnosti využiť pri vývoji Silverlight aplikácií.
- Podpora technológie .NET Framework s Rich Base Class Library: Kompatibilná časť kompletného .NET Frameworku.
- Výkonné ovládacie prvky: V základnom prostredí sú k dispozícii prvky DataGrid, ListBox, Slider, ScrollViewer, Calendar,...
- Pokročilá podpora šablón: Uľahčuje úpravu vzhľadu a ovládanie aplikácie.
- Veľké priblíženie (Deep Zoom): Prináša interaktivitu pri prezeraní obrázkov v extrémne vysokých rozlíšeníach.
- Rozsiahla podpora sieťových služieb: Podpora REST, WS\*/SOAP, POX, RSS, a štandardných HTTP služieb.

Architektúra Silverlightu má 2 časti: Prezentačnú vrstvu, ktorá obsahuje komponenty a služby určené na generovanie používateľského rozhrania a interakciu s používateľom. Používateľské rozhranie môže využívať renderovanie vektorovej a bitmapovej grafiky, textový výstup, animácie a prezentáciu multimediálneho obsahu vo formáte WMA, VC1 a MP3. Interakcia s užívateľom zahŕňa obsluhu udalostí generovaných používateľom pomocou myši a klávesnice. Na úroveň tejto vrstvy môžeme priradiť aj DOM API (Document Object Model) pre Silverlight a XAML. Druhou vrstvou je inštalačný a aktualizovaný komponent pre internetový prehliadač.



Obr. 6: Architektúra Silverlightu pre verzie 1.0 a 2

## 2.6.2. Multiplatformovosť a perspektíva

Z hľadiska implementácie je Silverlight multiplatformový plugin do prehliadača. Samotný plugin má veľkosť okolo 4,5 MB a inštalácia do prehliadača trvá necelých 30s a väčšinou aj bez nutnosti následného reštartovania prehliadača.

Ako vidieť v tabuľke, Silverlight je už podporovaný na väčšine hlavných prehliadačoch. Chýba prehliadač Opera, ktorá neoficiálne podporuje Silverlight. Súčasný stav je však taký, že na fórach sa objavujú sťažnosti na občasnú nefunkčnosť pluginu v tomto prehliadači.



	Internet Explorer 8	Internet Explorer 7	Internet Explorer 6	FireFox 3	FireFox 2	FireFox 1.5	Safari
Windows Vista	Áno	Áno	x	Áno	Áno	Áno	x
Windows XP SP2	Áno	Áno	Áno	Áno	Áno	Áno	x
Windows 2000	x	x	Áno**	x	x	x	x
Windows Server 2003 (okrem IA-64)	Áno	Áno	Áno	Áno	Áno	Áno	x
Mac OS 10.4.8+ (PowerPC)	x	x	x	x	Áno*	Áno*	Áno*
Mac OS 10.4.8+ (Intel-based)	x	x	x	x	Áno	Áno	Áno

\* Ien Silverlight 1.0; \*\* Ien Silverlight 2

Čo sa týka kompatibility s operačnými systémami, z tabuľky je vidieť podporu ako pre Windows tak aj pre Mac OS. Plugin pre Linux a ostatné platformy založené na Unix/X11 vyvíja spoločnosť Novell a projekt dostal názov Moonlight. Momentálne je finálna verzia Moonlight 1.0, ktorá je ekvivalentná k Silverlight 1.0, pričom v septembri 2009 je naplánovaný príchod Moonlight 2.0, ktorý je ekvivalent k Silverlight 2.

Dôležitá otázka je, či má Silverlight vôbec perspektívu. Je to nová technológia konkurujúca Adobe Flashu a Adobe Flexu, pričom najmä Flash má neotrasiteľné postavenie na trhu. Tieto technológie v princípe nepotrebujú žiaden ďalší plugin do prehliadača, keďže momentálne 99% PC pripojených na internet ma nainštalovaný Adobe Flash Player.

Zdá sa však, že Silverlight má perspektívu a oproti Flashu a Flexu má niekoľko výhod, ako napríklad podpora .NET jazykov. Môžeme si teda svoju Silverlight web aplikáciu napísať kompletne napríklad v C#, alebo Visual Basicu, čo už sú silné programovacie jazyky a vo svete rozšírené. Oproti tomu ActionScript je špecifický programovací jazyk navrhnutý výhradne pre Flash a Flex. Stále sa však môže zdať, že to je síce pekné, ale takmer každý počítač, ktorý je pripojený k internetu ma nainštalovaný Adobe Flash Player, ako je teda na tom Silverlight? Keďže za týmto projektom stojí Microsoft, čo je obrovský nadnárodný gigant, tak má samozrejme prostriedky ľudské aj finančné na spropagovanie tejto technológie. Do technológie, v ktorej vidí budúcnosť na webe.

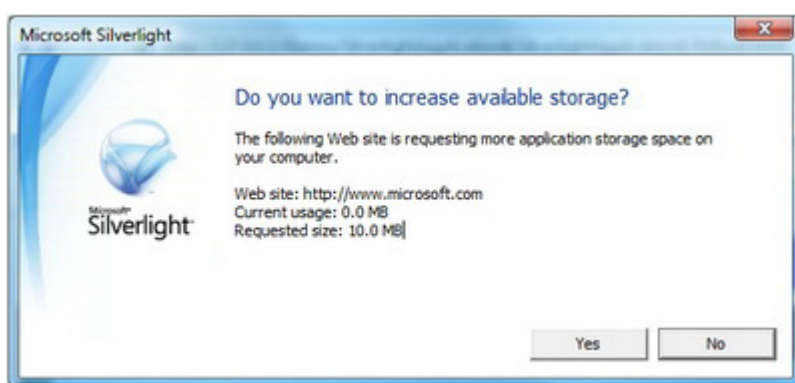
Podarilo sa mu napríklad dosiahnuť, aby internetové videoprenosy z Olympijských hier v Pekingu boli založené na tejto platforme. Taktiež záujem o Silverlight potvrdil Fínsky výrobca telefónov NOKIA svojim vyhlásením, že Silverlight bude podporovať na S60 so Symbian operačným systémom a v sade zariadení zo série 40 a taktiež na Nokia internet Tabletach.

Microsoft vkladá do tejto technológie (a technológie WPF) veľké nádeje, keďže iba relatívne nedávno bol predstavený Silverlight 2 a už je plánované predstavenie Silverlight 3, ktoré má priniesť množstvo noviniek, ako podporu video kodeku H.264, hardwarovú akceleráciu videa, trojrozmernú grafiku, ako aj lepšiu spoluprácu s Visual Studiom a bezplatným vývojovým prostredím Visual Web Developer Express.

### 2.6.3. Bezpečnosť

Silverlight 2.0 aplikácie bežia v bezpečnostnej obálke tzv. Sandbox-e. Vývojári nemajú možnosť nijako ovplyvniť tento Sandbox na klientskej strane, takže nemôžu prekonať obmedzenie a dostať sa so svojím kódom mimo túto vymedzenú bezpečnostnú obálku.

Sandbox poskytuje na klientskej strane aj lokálne úložisko dát (Local isolated storage) s kapacitou približne 10 MB [9], dialógové okno na otváranie súborov na klientskom PC a podobne. Je možné vyberať naraz aj viac súborov a tieto následne odoslať na server alebo uložiť do lokálneho úložiska. Ak niektorá aplikácia potrebuje viac miesta, tak sa zobrazí dialógové okno, ktoré nás vyzve na zvýšenie kapacity. Samozrejme to akceptujeme, iba ak takejto aplikácii dôverujeme.



Obr.7: Výzva k zväčšeniu Isolated Storage

Obsah súborov v lokálnom úložisku môžeme pomocou aplikačnej logiky meniť, no pôvodný súbor zostáva za každých okolností nezmenený a Silverlight aplikácia pracuje len s jeho kópiou. Ostatný diskový priestor mimo Local isolated storage je pre Silverlight aplikáciu nedostupný. Pomocou súborov s politikou (policy-files) môžeme kontrolovať všetky cross-domain http(s) požiadavky, teda požiadavky na prenos údajov z iných domén.

### 2.6.4. XAML

XAML je nový druh značkovacieho jazyka určeného na tvorbu aplikačného rozhrania pre WPF a Silverlight aplikácie. Vychádza z veľmi populárneho formátu XML. Na rozdiel od iných technológií, ktoré posielajú prehliadaču binárny obsah (napríklad Java Applety, ActiveX a Flash) a môžu mať problémy s firewall-mi a nekompatibilitou operačných systémov a prehliadačov, XAML je textový formát.

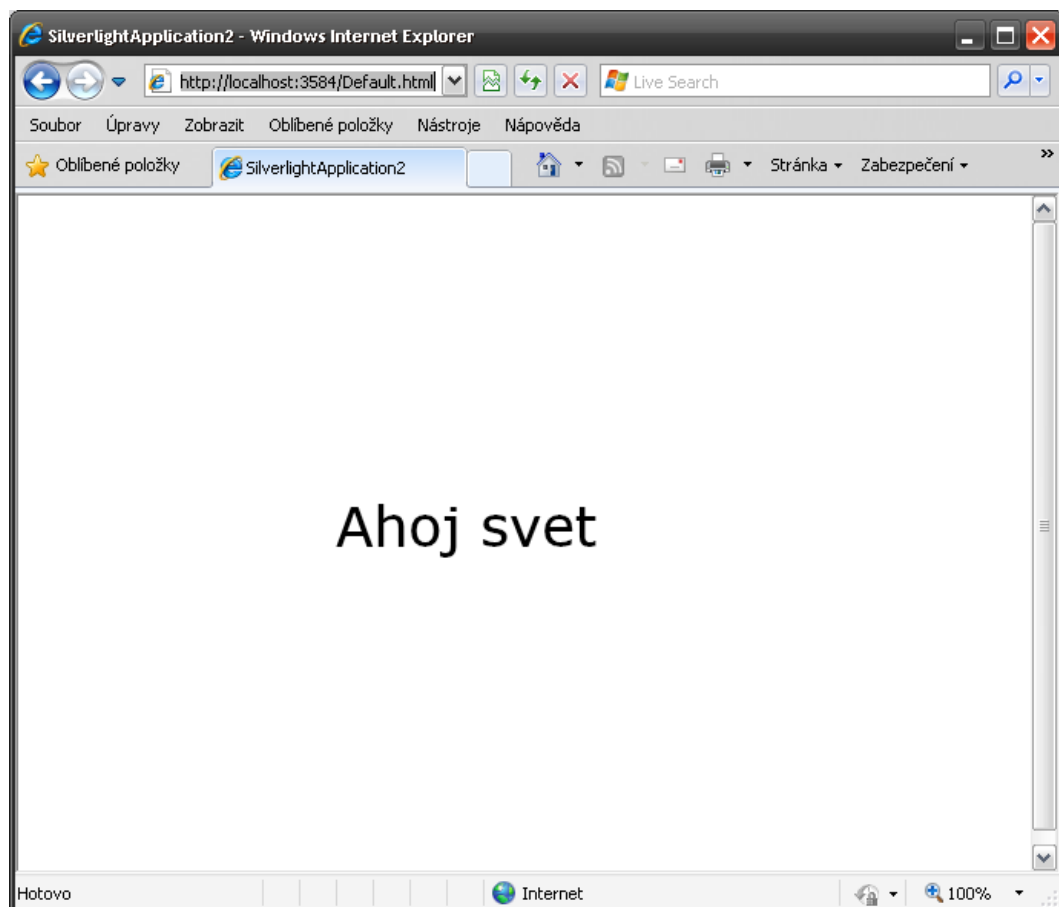
```

<UserControl
  xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
  x:Class="SilverlightApplication2.Page"
  Width="640" Height="480">

  <Grid x:Name="LayoutRoot" Background="White">
    <TextBlock Margin="207,190,248,227" TextWrapping="Wrap"><Run FontSize="36" Text="Ahoj svet"/>
    </TextBlock>
  </Grid>
</UserControl>

```

Obr.8: Ukážka jednoduchého XAML kódu



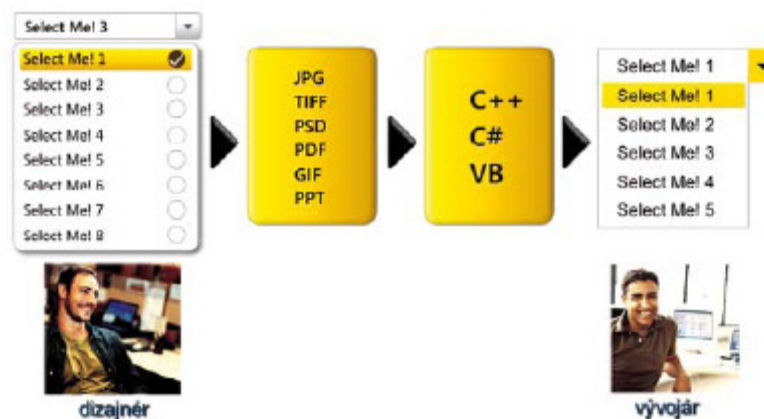
obr.9: A výsledok zobrazený v prehliadači

XAML je taktiež veľmi výkonný prostriedok pre tímový vývoj. Všetky moderné vývojárske nástroje podporujú možnosť tímovej spolupráce pri tvorbe softvérových projektov ako jeden z hlavných pilierov svojej koncepcie. Ukážeme výhody XAML pri spolupráci grafického designera a programátora.



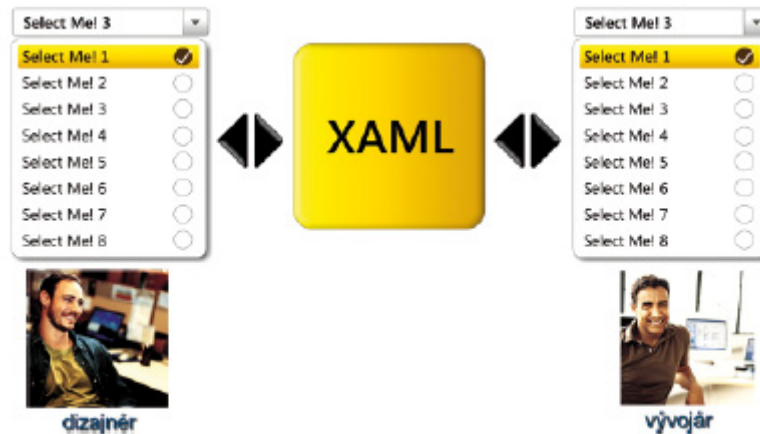
Obr.10: Timový vývoj

Designer predloží vývojárovi finálny návrh, ako by mala aplikácia vyzerat', vývojár začne s týmto návrhom pracovať, pričom sa pri implementácii tohto návrhu riadi možnosťami návrhového prostredia, v ktorom pracuje. Preto podoba niektorých ovládacích prvkov sa môže v konečnom dôsledku líšiť od pôvodného návrhu dodaného designerom.



Obr.11: Klasická spolupráca medzi designerom a vývojárom

Ak ale designer vytvorí svoj návrh vo formáte, ktorý je podporovaný vývojovým prostredím vývojára, tak tým docielime, že výsledná aplikácia bude vyzerat' rovnako ako návrh od designera a nebude sa v ničom líšiť. Vývojár jednoducho dostane od grafika návrh v XAML formáte a on doňho následne iba doprogramuje funkcionality.



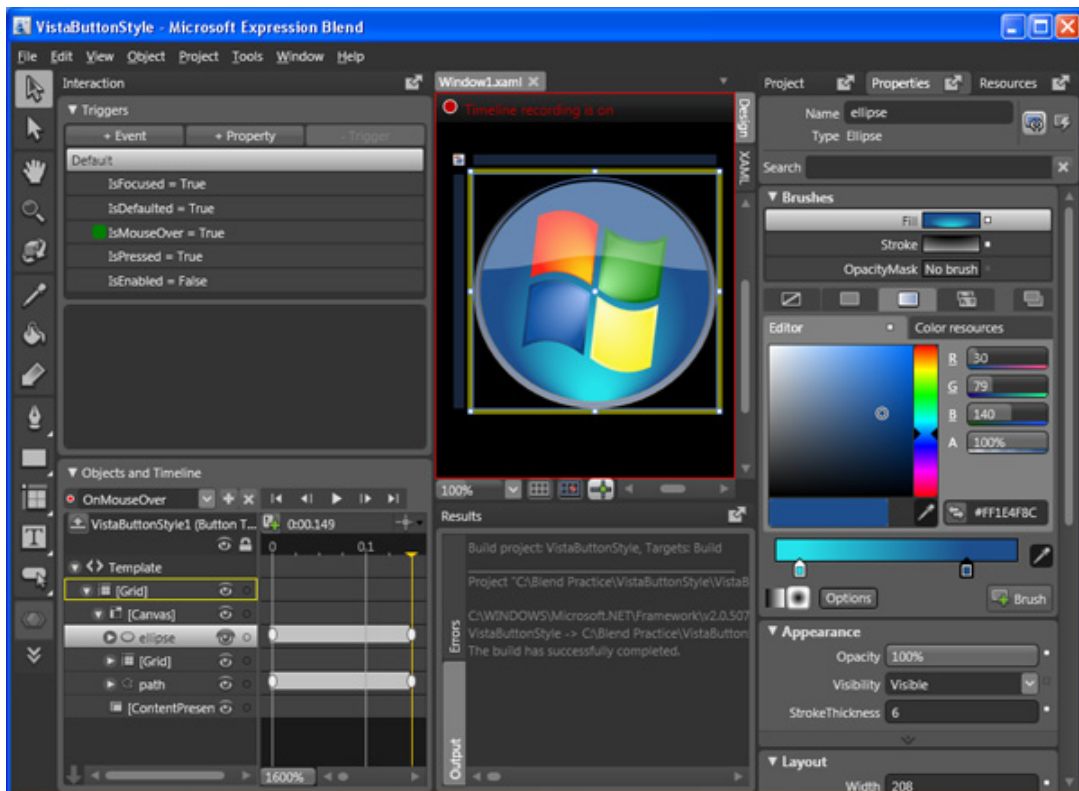
Obr.12: Spolupráca medzi designerom a vývojárom použitím XAML

V akom programe ale môže grafik navrhnuť svoj design a v akom programe mu zasa pridá vývojár funkcionality?

Microsoft pri príležitosti vydania Silverlightu ako novej technológie prišiel aj s novou komplexnou produktovou sadou Expression (momentálne verzia 2), ktorá obsahuje produkty: Microsoft Expression Web, Microsoft Expression Blend, Microsoft Expression Design, Microsoft Expression Media a Microsoft Expression Encoder.

Asi najzaujímavejším programom z Expression rady je Expression Blend. Tento program je flexibilné grafické prostredie, ktoré pomáha pri tvorbe aplikácií s interaktívnou podporou 3D zobrazovania a prehrávania multimédií. Umožňuje vytvorenie a úpravu prezentačnej vrstvy webových aplikácií pričom používa XAML kód. Takže pomocou tohto vizuálne veľmi prívetivého a prepracovaného programu sa dá doslova nakresliť efektná aplikácia a výstupom bude XAML kód, ktorý následne vývojár oživí.

Programy z edície Expression sú platené, ale na druhej strane XAML súbor sa dá editovať aj napríklad v obyčajnom poznámkovom bloku. Nie je to síce komfortné, ale oproti Flashu to poskytuje aspoň nejakú možnosť editácie.



Obr.13: Uživatelské prostredie programu Microsoft Expression Blend

Expression Blend slúži hlavne potrebám designerov na jednoduchú a efektívnu prácu so XAML súbormi. Vývojári sú na tom čo sa týka možnosti práce so Silverlightom o niečo lepšie. Môžu využiť buď platené riešenie Visual Studio 2008 so SP1, alebo zdarma dostupný software Visual Web Developer Express taktiež so SP1. Nad tieto vývojové prostredia treba ešte doinštalovať balíček „Silverlight Tools for Visual Studio 2008 SP1“, po ktorého nainštalovaní sa zobrazí nový typ projektu „Silverlight Application“. Tieto programy sú navzájom úzko prepojené s Expression Blendom, pričom sa dá plynule prechádzať z editácie XAML kódu v Blende do týchto programov a naopak.



Obr.14: Prepojenosť návrhových a vývojových programov

## 3. Dostupné riešenia

V tejto kapitole sú predstavené niektoré zaujímavé riešenia, ktoré slúžia na vykresľovanie grafov na internete. Prípadne komplexnejšie riešenia, ktorých iba časť ma za úlohu prácu s grafmi. Sú ukázané ich prednosti, nedostatky. Pridaný je aj komplexnejší zoznam ďalších riešení určených na kreslenie grafov s jednoduchým opisom funkcionality. Nie je cieľom podrobne popísať všetky riešenia, keďže to by stačilo na samotnú diplomovú prácu, ale iba predstaviť niektoré najzaujímavejšie.

### 3.1. GigaSoft ProEssentials Charts

**www:** [www.gigasoft.com](http://www.gigasoft.com)

**licencia:** od 650\$

Toto riešenie ponúka široké možnosti vykresľovania grafov ako vo WinForm tak aj WebForm. Je dodávané ako rozhranie pre ActiveX, .NET (Windows Forms), ASP.NET (Web Forms), DLL, a VCL.

.NET (WinForm)	VB.Net	C#.Net	C++.Net
ASP.NET (WebForm)	VB Asp.Net	C# Asp.Net	
ActiveX	VB6	Asp IIS	
DLL	Win32		
VCL	Delphi		

Obr.15: Dodávané rozhrania

Grafy určené pre zobrazovanie na internetových stránkach sa môžu vykresliť ako obrázky, prípadne ako ActiveX. ActiveX podporuje zaujímavé možnosti interaktivity čo sa týka zobrazovania iba určitého úseku dát, taktiež je príjemná možnosť prepnúť grafy na celú obrazovku.

Ponúka množstvo neštandardných typov grafov, ktoré iné riešenia neponúkajú, avšak neumožňuje vykresľovať grafy ako Flash. Keďže voči ActiveX panujú predsudky a taktiež povoľovanie prehrávania ActiveX prvku v prehliadači môže ľudí otravovať, možno to považovať za nedostatok.

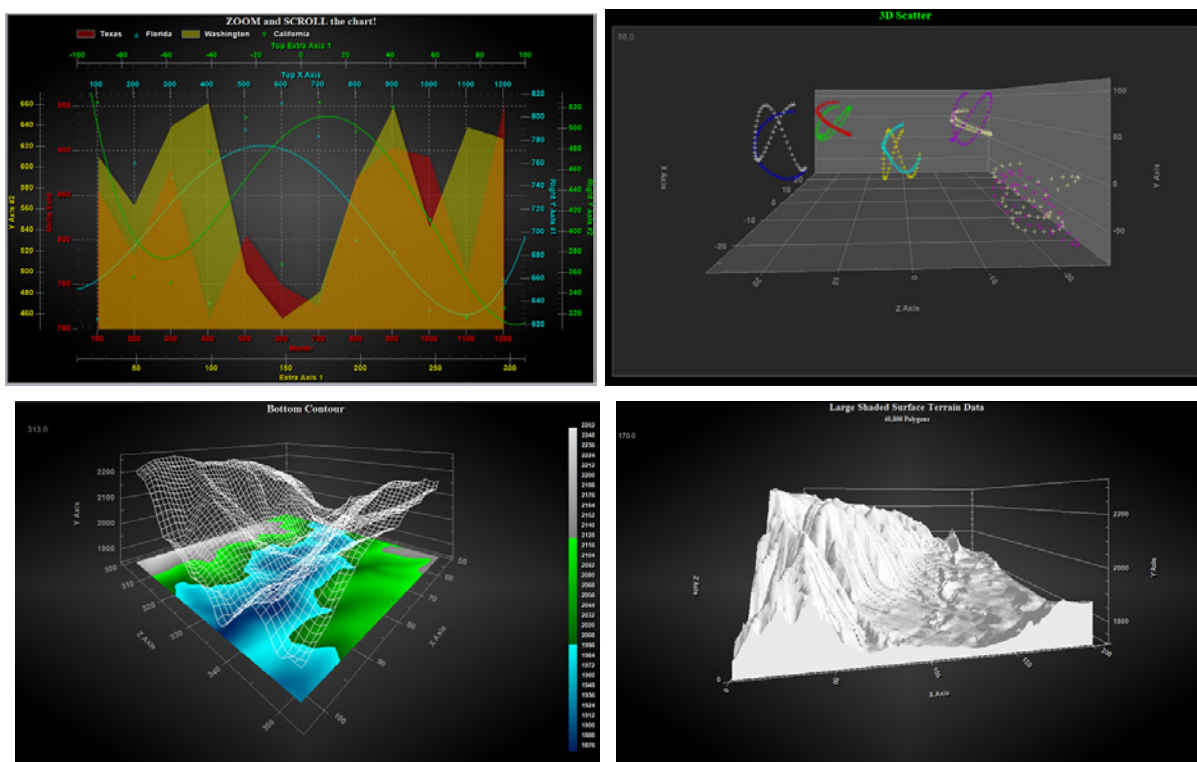
Čo sa týka vytvorenia takéhoto grafu, ide hlavne o programátorskú záležitosť, keďže grafy treba „naprogramovať“ v niektorom z .NET jazykov za použitia grafovej komponenty do Visual Studio.

Riešenie ProEssentials je kompatibilné s Microsoft Visual Studio.NET 2005, 2003, a aj 2002, ďalej s Visual Studio 6.0, Visual Basic 6.0 a Delphi/Builder.



Toto vylučuje z hry použitie týchto grafov napríklad na PHP stránkach, alebo do klasických HTML stránok. Je síce možné vygenerovať si takýto graf a ten následne si vložiť na svoje stránky, ale je to veľmi neefektívne riešenie.

Toto riešenie je vhodné pri tvorbe špecifického internetového/intranetového portálu založeného na technológii .NET so špecifickými typmi grafov.



Obr.16: Niektoré zaujímavé typy grafov

## 3.2. Telerik RadChart for ASP.NET AJAX

**www:** [www.telerik.com](http://www.telerik.com)

**licencia:** od 799\$

Telerik RadChart for ASP.NET AJAX je súčasťou väčšieho balíka RadControls for ASP.NET AJAX, ktorý obsahuje množstvo ďalších komponentov ako napríklad Kalendár, TabStrip, Slider a mnoho ďalších. Celkovo je v tomto balíku 27 hlavných a 35 pomocných ovládacích prvkov, s ktorých pomocou sa dajú vytvoriť naozaj veľmi elegantné užívateľské rozhrania. Čo sa týka grafického spracovania, dali si na všetkých prvkoch naozaj veľmi záležať.

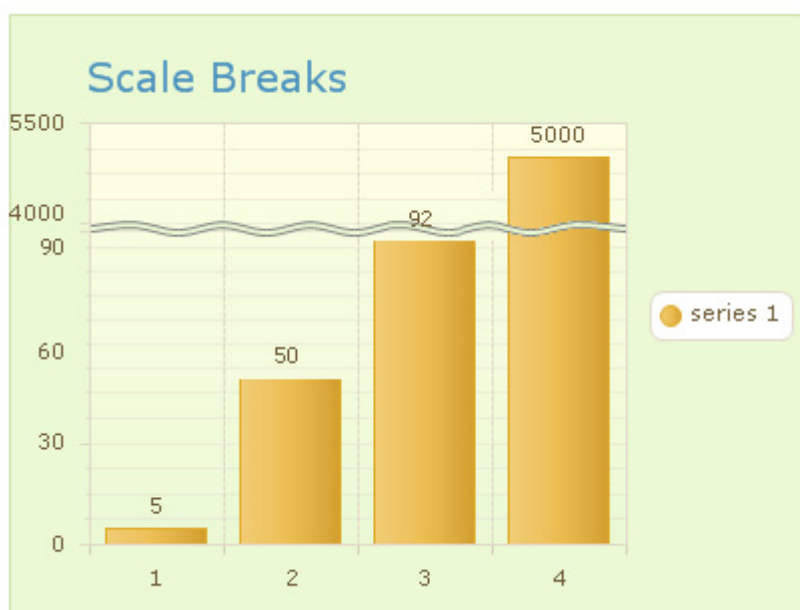
Ako názov napovedá, ide o ASP.NET ovládacie prvky, takže taktiež ako v predošlom riešení nie je možné zobrazovať grafy na inej technológii ako ASP.NET. Tým sa znižuje okruh potenciálnych



zákazníkov. Hlavná výhoda týchto grafov, ako aj všetkých ovládacích prvkov je ich naozaj veľmi efektívne grafické spracovanie a všeobecná funkčnosť.

Čo sa týka grafov, sklamaním je, že sú generované iba ako obrázky, teda možnosť interaktivity je veľmi malá a každý pokus o interaktivitu vedie ku komunikácii so serverom a vygenerovaním nového obrázka. Zaujímavá je však široká podpora dátových zdrojov ako LINQ, LinqDataSource, ADO.NET DataServices, MVC Dynamic Data, EntityDataSource.

Taktiež je možné zapnúť ScaleBreak, ktorého význam najlepšie opisuje nasledujúci obrázok.



Obr.17: Graf so zapnutým ScaleBreaks

Zjednodušene povedané, ScaleBreak vizuálne zmenší hodnotové rozdiely medzi stĺpcami (v tomto prípade), pokiaľ jedna alebo viacej hodnôt sú veľmi veľké a ostatné sa pri nich strácajú na X-ovej osi.

Celkovo sa po grafickej stránke jedná o veľmi prepracovaný komponent, ktorý ponúka zaujímavú funkčnosť ako ZOOM, ScaleBreak, využitie LINQ ako dátového zdroja. Avšak po stránke funkčnej je možnosť interaktivity grafov, vďaka zobrazovaniu iba ako obrázkov malá a obmedzená na ASP.NET aplikácie.

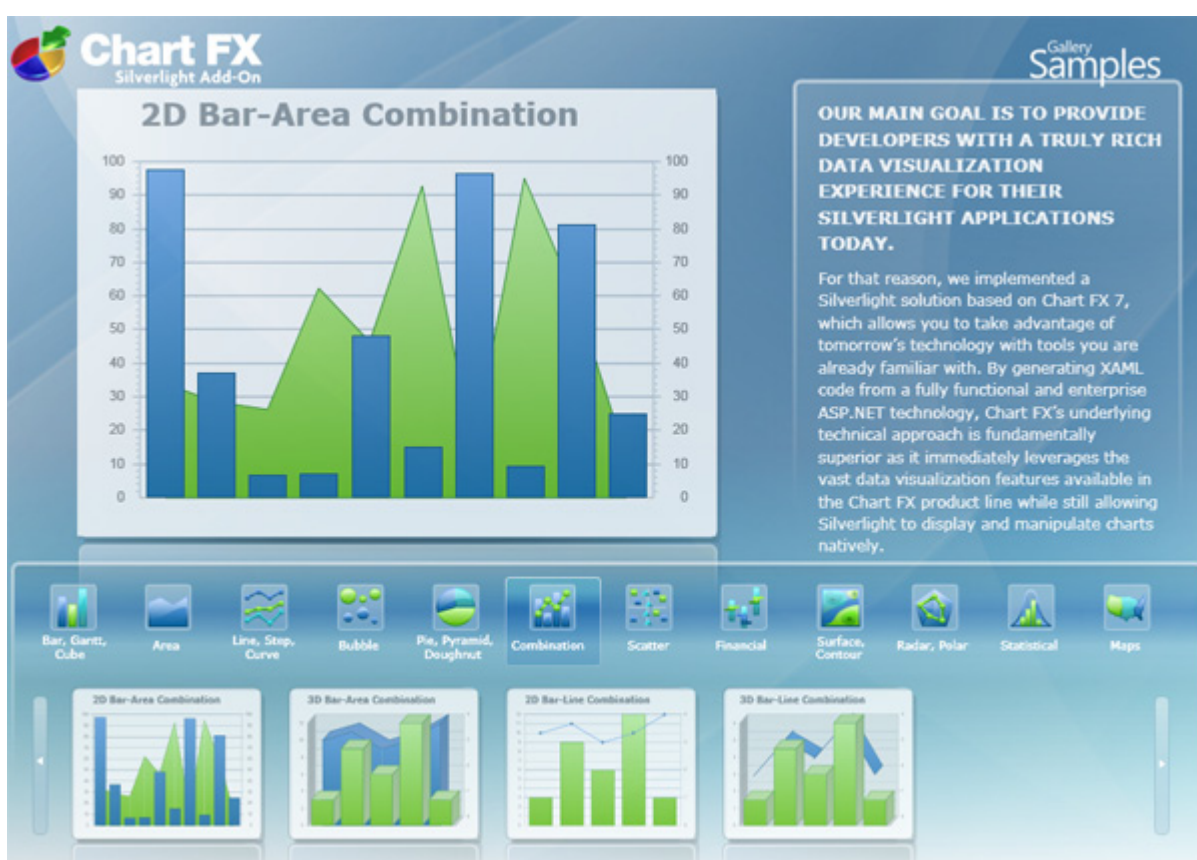
### 3.3. Chart FX

www: [www.softwarefx.com](http://www.softwarefx.com)

licencia: od 850\$

Čo sa týka možností použitia Chart FX tak tie sú vcelku rozsiahle. Dá sa integrovať do Visual Studio .NET (2003), ako aj do Visual Studio 2005 a Visual Studio 2008. Vie zobrazovať Grafy v Silverlighte ako aj v WPF. Chart FX je dostupný taktiež pre Javu, ASP, Client Server & SQL Reporting aplikácie.

Zobrazovania grafov pomocou technológie Silverlight je na veľmi nízkej úrovni. Chart FX Vie zobrazovať množstvo typov grafov, ale interaktivita je naozaj malá, ak nie nulová.



Obr.18: Chart FX a jeho Silverlight grafy

## 3.4. Fusion Charts

**www:** [www.fusioncharts.com](http://www.fusioncharts.com)

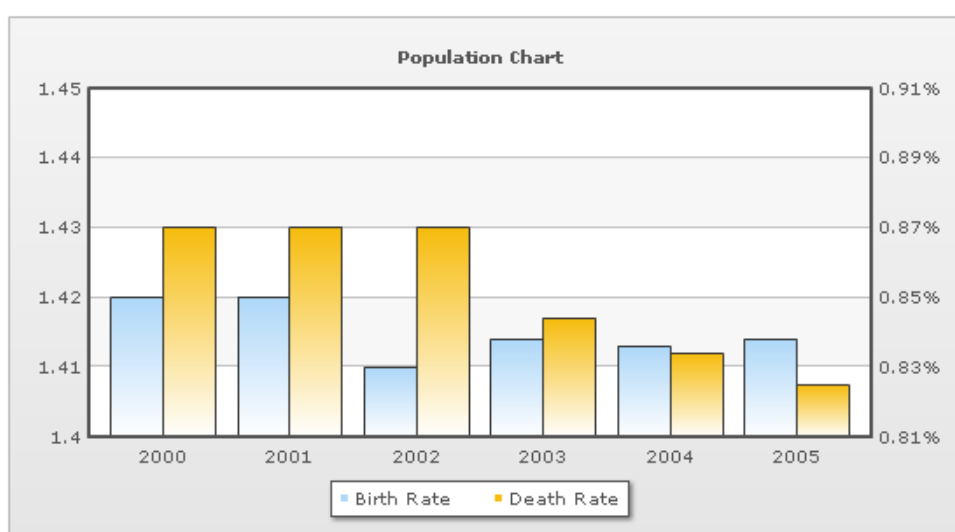
**licencia:** od 500\$

Pomocou Fusion Charts grafov, momentálne vo verzii 3 sa dajú ľahko generovať FLASH grafy pre webové a „desktopové“ aplikácie. Toto riešenie sa výzorom grafov blíži produktu Visifire (ktorý je spomínaný neskôr), avšak je omnoho komplexnejšie. Fusion Charts Free je chudobnejšia, neplatená verzia systému Fusion Charts v3.

Je ponúkaných 45 typov rôznych typov grafov, ktoré môžu byť upravovateľne cez XML API Fusion Charts, Preto nie je potrebná žiadna znalosť Flash technológie. Generované grafy sú animované, veľmi efektné a poskytujú aj niektoré zaujímavé interaktívne funkcie, ako priama editácia dát v grafe jednoduchým ňaťahovaním stĺpcov, export do PDF, obrázkov. Taktiež je možnosť integrácie s JavaScriptom. Zaujímavá je možnosť s pomocou integrácie JavaScriptu (AJAX) vytvoriť komunikáciu medzi grafmi, kedy vyradenie hodnoty v jednom grafe, spôsobí zmenu v inom, ako ukazuje jedna z pokročilých ukážok na stránke tohto produktu.

Inštalácia spočíva v stiahnutí SWF „zdrojových“ súborov, vytvorenia XML s dátami a nastaveniami a následným vložením odkazu do svojich stránok. Použitie týchto grafov je naozaj široké. Môžu byť integrované napríklad pomocou technológií ASP, ASP.NET, PHP, JSP, ColdFusion, Ruby on Rails, Python, alebo klasických HTML stránok.

Taktiež Fusion Charts pracujú so všetkými databázami ako MS SQL, Oracle, MySQL, PostgreSQL, CSV.



Obr.19: Ukážka Fusion Charts grafu

## 3.5. Dundas Chart

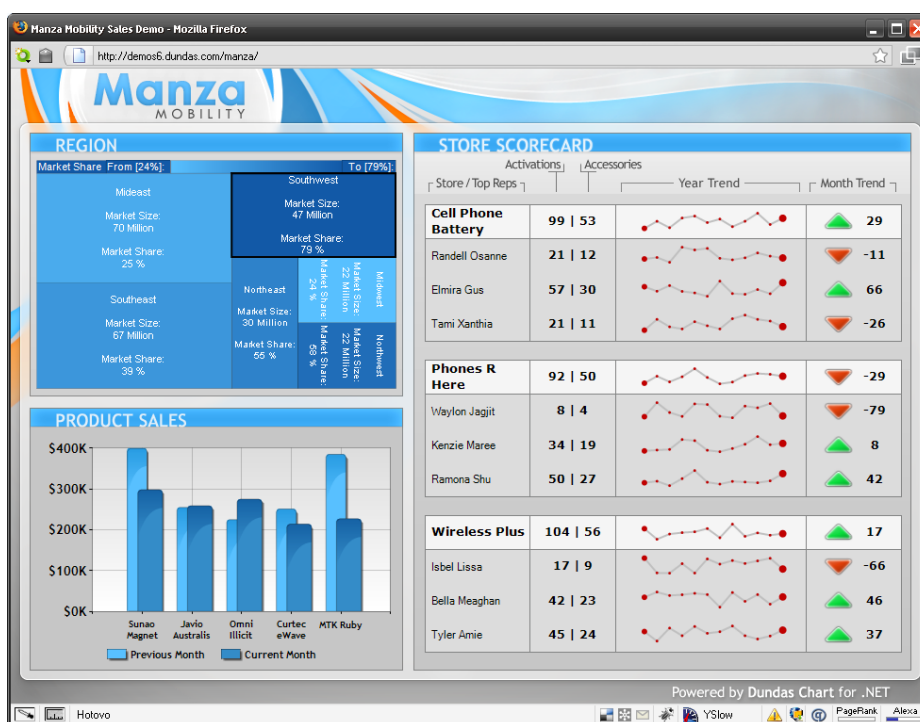
www: [www.dundas.com](http://www.dundas.com)

licencia: od 700\$

Silný nástroj, rovnako ako Fusion Charts, avšak Dundas je určený pre .NET, Share Point a Reporting Services. Momentálne sa testuje verzia Dundas Chart pre Silverlight, ktorá je v štádiu Beta.

Zaujímavosťou je, že Microsoft nedávnou akvizíciou tejto firmy uvoľnil zadarmo grafický ovládací prvok na generovanie grafov do webových aplikácií (ASP.NET), ktorý je založený na technológii Dundas Chart. Tento ovládací prvok bude aj súčasťou najnovšieho .NET Frameworku. Ako sme mali možnosť vyskúšať, tento prvok vie momentálne generovať grafy iba vo forme obrázkov.

Rovnako ako Fusion Charts grafy aj Dundas podporuje veľké množstvo typov grafov od štandardných až po menej štandardné. Samotné grafy sú vykresľované ako bitmapa, prípadne ako Flash, alebo SVG. Toto riešenie patrí spolu Fusion Charts medzi najkomplexnejšie riešenia pre vykresľovanie grafov na webe.



Obr.20: Ukážka ako môže vyzerat' grafy Dundas Chart

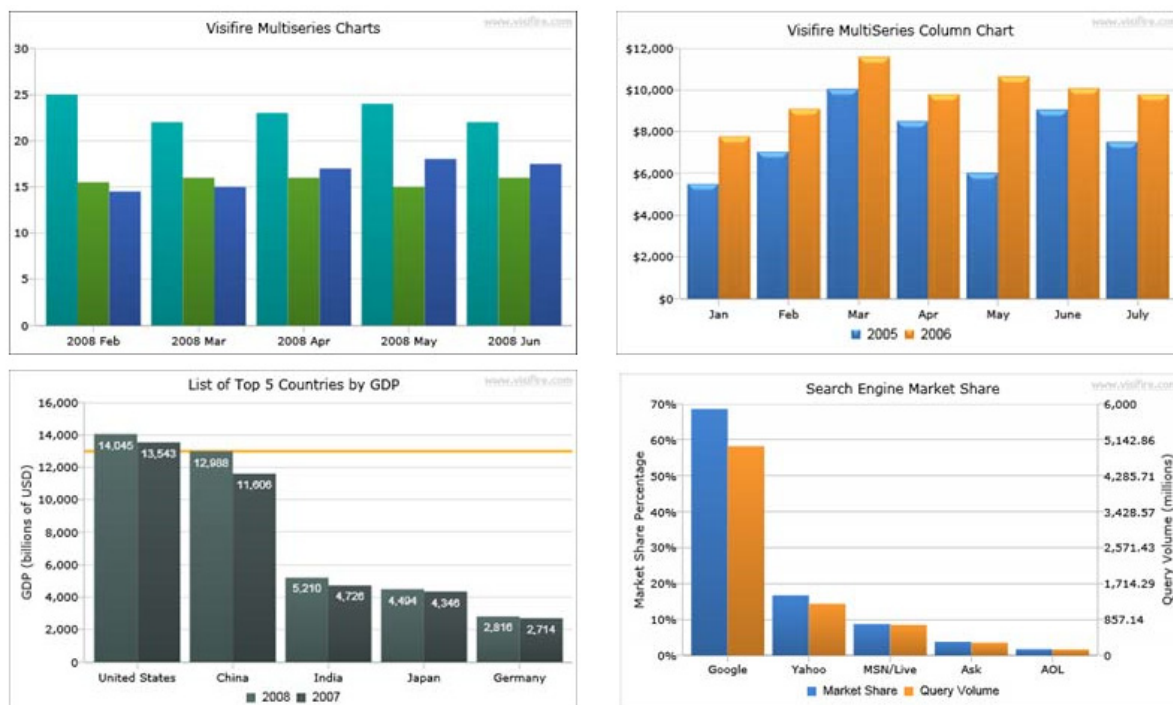
## 3.6. Visifire

www: [www.visifire.com](http://www.visifire.com)

licencia: GPL 3.0

Visifire [14] je „open source“ sada komponentov určených na zobrazovanie grafov pomocou Silverlightu a WPF. Všetky grafy sú animované a ich vytvorenie zaberie minimum času aj začiatočníkovi. Nevýhodou je, že podporujú iba jednoduché zobrazenie grafu, pričom interaktivita je len minimálna. Prípadne rozšíriteľná cez JavaScript.

Visifire je jednoducho implementovateľný na stránky, nezávislý na použitej serverovej technológii. Môže byť použitý s ASP, ASP.Net, PHP, JSP, ColdFusion, Ruby on Rails, alebo obyčajným HTML.



Obr.21: Ukážka Visifire grafov

## 3.7. Rozšírený prehľad riešení

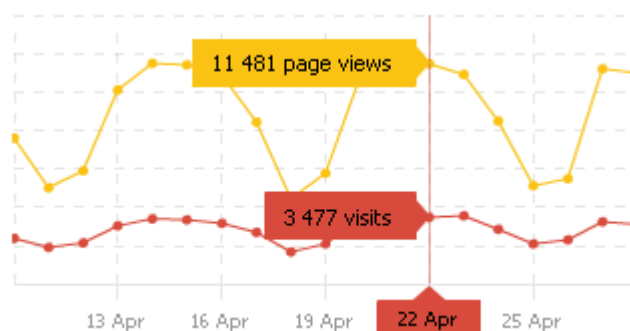
V tejto kapitole je predstavený rozšírený zoznam [12] ďalších riešení, ktoré umožňujú vykresľovať grafy na webe. Zoznam je rozdelený na sekcie podľa technológie, akou vykresľujú grafy na stránky. Kvôli rozsahu zoznamu sú riešenia popísané iba v skrátenej forme, pričom sú predstavené ich hlavné výhody.

### 3.7.1. Flash

**amCharts** ([www.amcharts.com](http://www.amcharts.com))

**licencia:** linkware, alebo od 85€

Jednoducho nastaviteľné grafy, ktoré sú generované za behu. Ponúkané sú však iba základné typy grafov, ktoré môžu byť 2D aj 3D. Dáta môžu byť načítavane z XML, prípadne CSV. Niektoré grafy majú animovaný efekt.

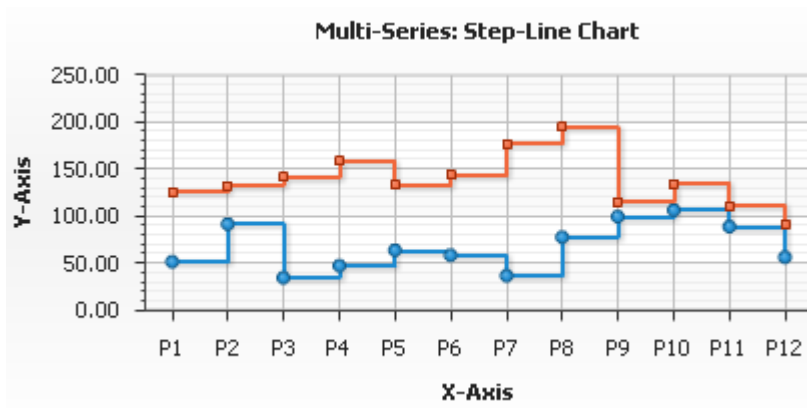


Obr.22: Ukážka amCharts grafu

**AnyChart Flash Chart Component** ([www.anychart.com/products/anychart/overview](http://www.anychart.com/products/anychart/overview))

**licencia:** 500\$

Komponent, ktorý dokáže generovať množstvo, prevažne základných typov grafov. Môže byť použitý s ASP.NET, ASP, PHP, JSP, ColdFusion, Ruby on Rails, jednoduché HTML a tiež v PowerPoint prezentáciách. Ako dátový zdroj používa XML súbor. Grafy sa dajú uložiť ako obrázok, napríklad pre prípadne poslanie ako prílohu v emaily.

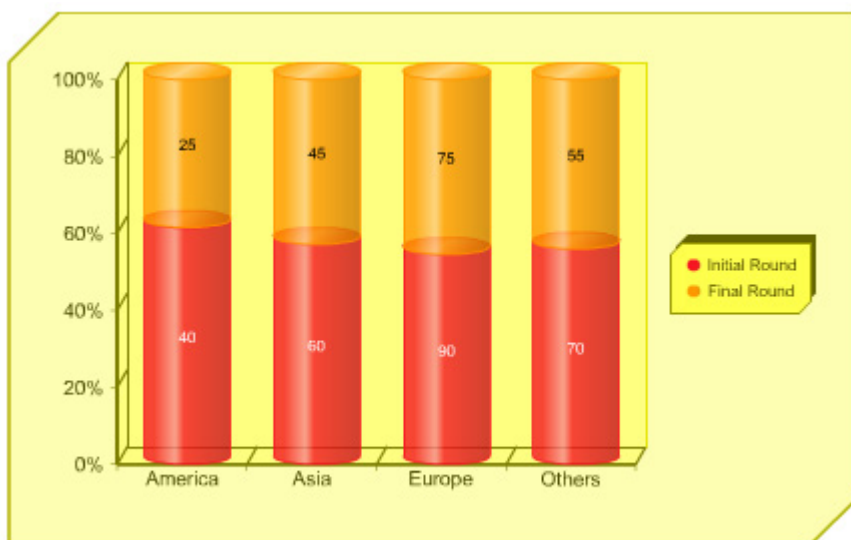


Obr.23: Ukážka AnyChart Flash Chart komponentu

**Rich Chart Server for .NET** ([www.blue-pacific.com/products/RichChartServer](http://www.blue-pacific.com/products/RichChartServer))

**licencia:** freeware, alebo od 5\$/deň

Vie generovať interaktívne Flash grafy z ASP.NET a Winf Forms aplikácii. Má flexibilný systém šablón na integrovanie dynamického obsahu z veľkého množstva dátových zdrojov , ktoré sú dostupné v .NET.

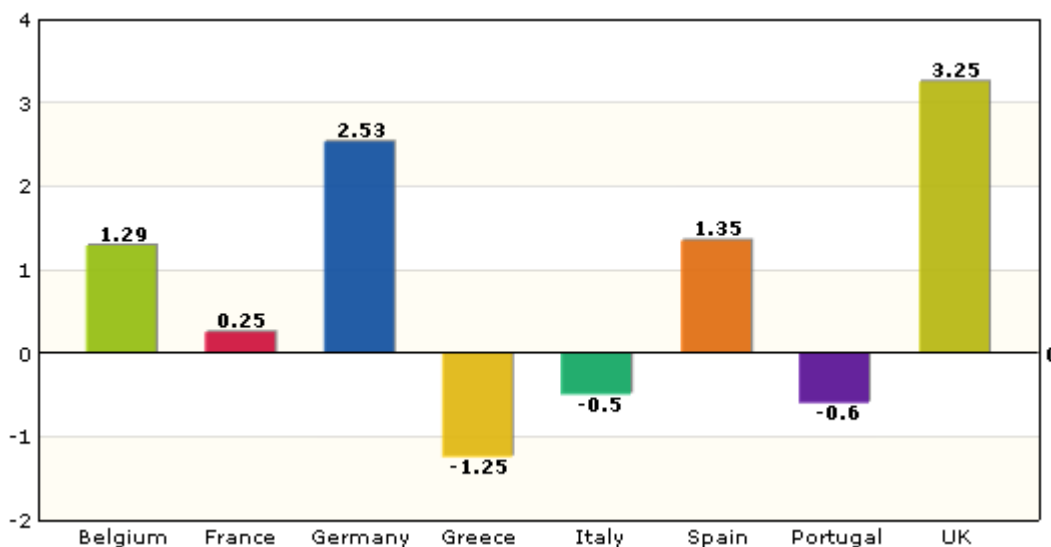


Obr.24: Ukážka Rich Chart Server for .NET grafu

**Flash Charts Pro** ([www.web-site-scripts.com/flash-charts](http://www.web-site-scripts.com/flash-charts))

**licencia:** od 60\$

Generuje okolo 30 typov grafov, ktoré sú rozdelené v 9 kategóriách. Nie je nutná žiadna inštalácia. Stačí stiahnuť Flash súbor a priložiť XML súbor, ktorý slúži ako konfiguračný a dátový zdroj. Podporuje multijazykovú podporu (UTF-8).

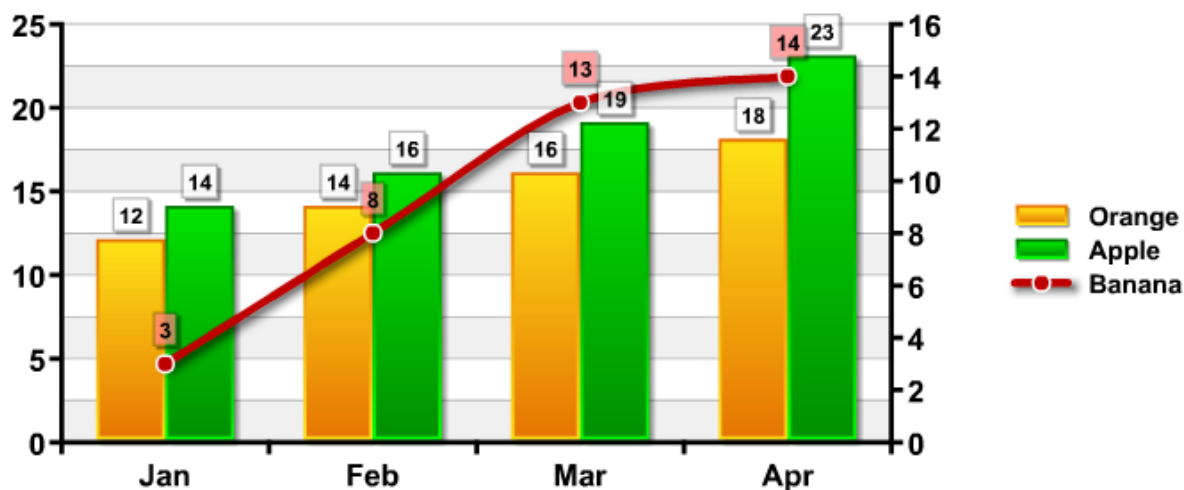


Obr.25: Ukážka Flash Charts Pro grafu

## Swift Chart Generator ([www.globfx.com/products/swfchartgen](http://www.globfx.com/products/swfchartgen))

licencia: od 75\$

Riešenie pre interaktívne Flash grafy. Pomocou XML súboru, ktorý môže byť samozrejme generovaný dynamicky v nejakom programovacom jazyku, popíšete vzhľad grafu, ako aj dáta a Flash súbor z nich vygeneruje graf.

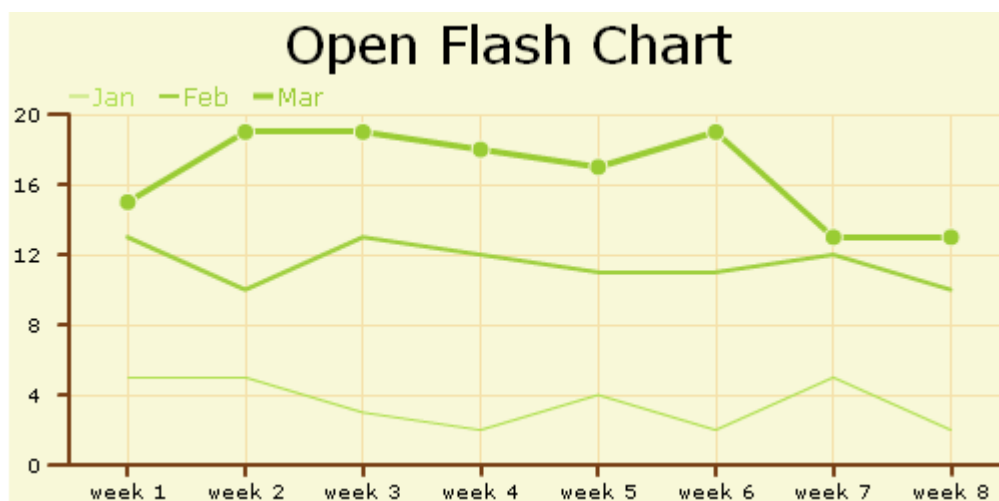


Obr.26: Ukážka Swift Chart Generator Pro grafu

## Open Flash Chart (OFC) ([www.teethgrinder.co.uk/open-flash-chart](http://www.teethgrinder.co.uk/open-flash-chart))

licencia: LGPL

Open-source projekt, ktorý ponúka 35 rôznych typov grafov s návodmi, ako sa pomocou skriptov dá rozšíriť funkcionalita. Dáta môžu byť uložené napríklad aj v obyčajnom .txt súbore. Je dostupných množstvo príkladov a ukážok.



Obr.27: Ukážka Open Flash Chart grafu

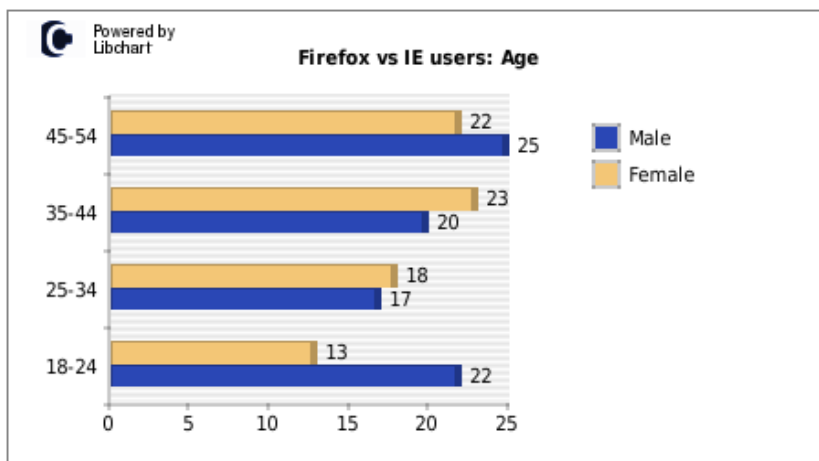


### 3.7.2. PHP a ASP knižnice

**Libchart** (<http://naku.dohcrew.com/libchart/pages/introduction/>)

**licencia:** freeware

Jednoduchá PHP knižnica, pomocou ktorej môžete zobrazovať čiarové, stĺpcové a koláčové grafy. Je distribuovaná pod licenciou GNU General Public License (GPL).

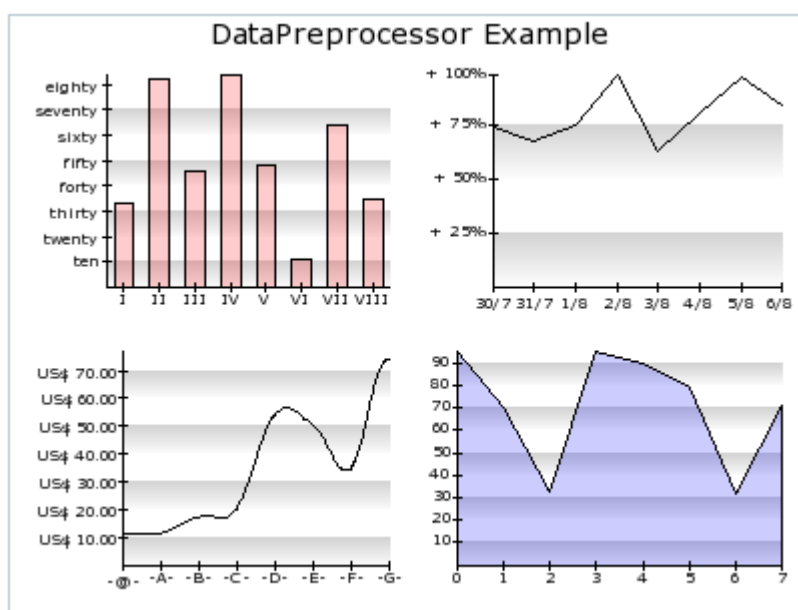


Obr.28: Ukážka Libchart grafu

**Image\_Graph** (<http://pear.veggerby.dk>)

**licencia:** freeware

Image\_Graph tiež známa ako GraPHPite ponúka 14 typov grafov, ktoré ponúkajú širokú možnosť nastavení a prispôbení k svojim potrebám. Na stránke sa nachádza cez 80 rôznych ukážok.

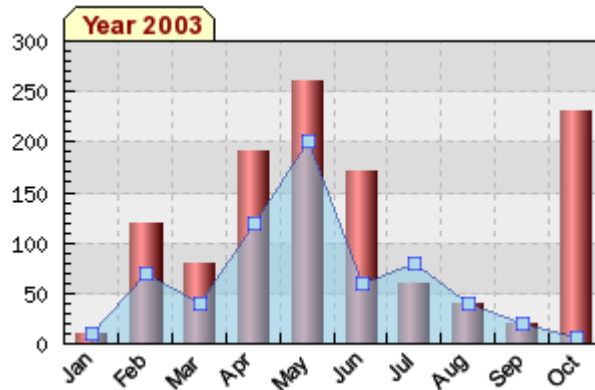


Obr.29: Ukážka Image\_Graph grafu

JpGraph ([www.aditus.nu/jpgraph](http://www.aditus.nu/jpgraph))

licencia: QPL 1.0

Objektovo orientovaná knižnica pre PHP ponúkajúca okolo 15 typov grafov (koláčové aj 3D).

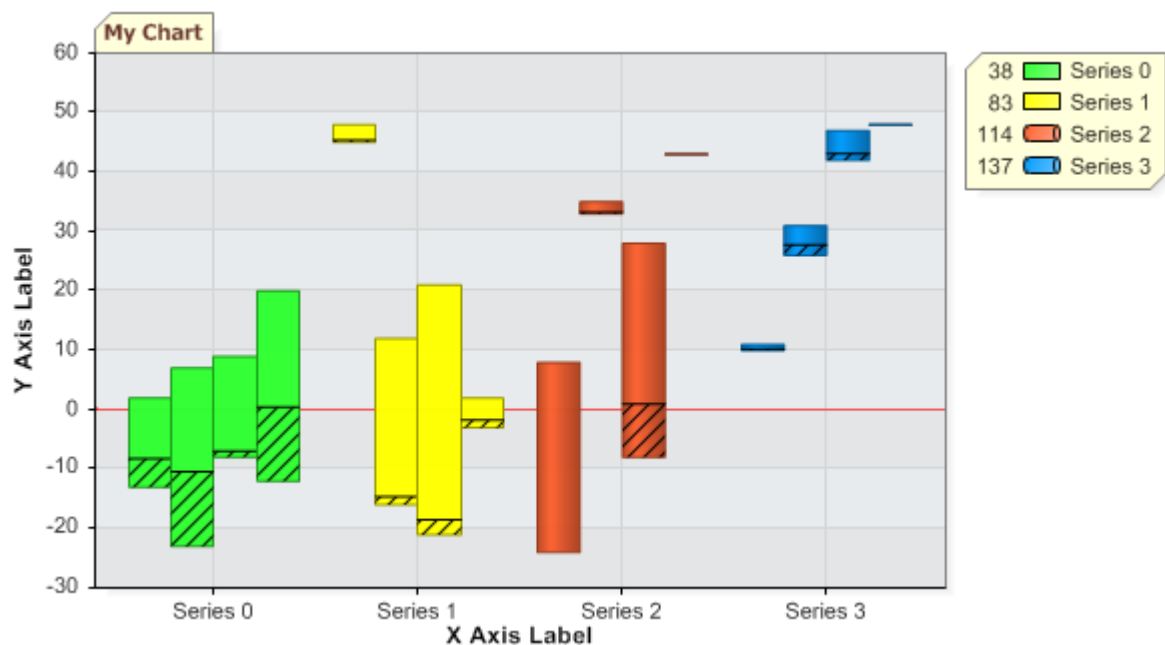


Obr.30: Ukážka JpGraph grafu

.netCHARTING ([www.dotnetcharting.com/overview.aspx](http://www.dotnetcharting.com/overview.aspx))

licencia: od 395\$

Využíva .NET framework a GDI+. Programátorom v C# a VB.NET pracujúcim s ASP.NET alebo WinForms ponúka veľké množstvo grafov (2D, 3D), ktoré sa dajú pomocou tohto riešenia zobrazovať.



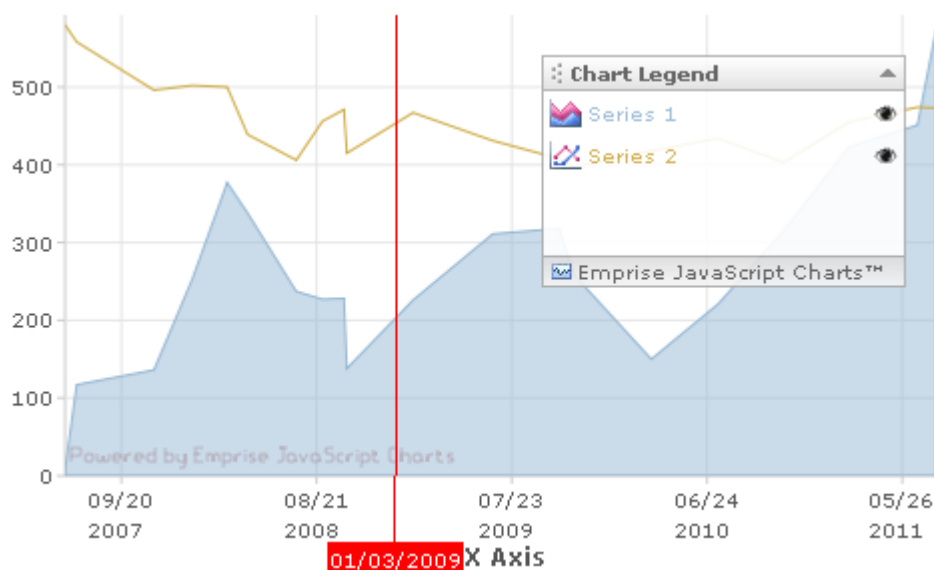
Obr.31: Ukážka .netCHARTING grafu

### 3.7.3. JavaScript

#### Emprise JavaScript Charts ([www.ejschart.com](http://www.ejschart.com))

**licencia:** free, alebo od 100\$

Riešenie založené na čistom JavaScripte. Podporuje interaktivitu, ako napríklad rôzne udalosti (mouse events,...), približovanie (ZOOM), skrolovanie. Podporuje dáta v XML formáte. Grafy sú plne prispôsobiteľne cez dobre zdokumentované premenné a metódy.

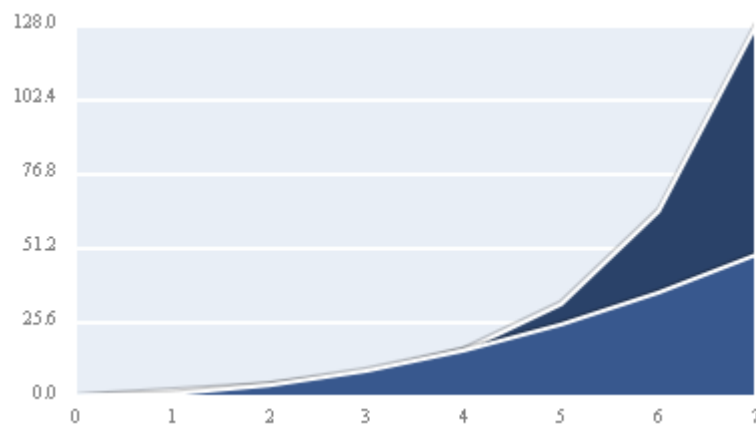


Obr.32: Ukážka Emprise JavaScript Charts grafu

#### PlotKit - Javascript Chart Plotting ([www.liquidx.net/plotkit](http://www.liquidx.net/plotkit))

**licencia:** BSD

Experimentálna JavaScriptová knižnica s podporou HTML tagu Canvas a SVG.

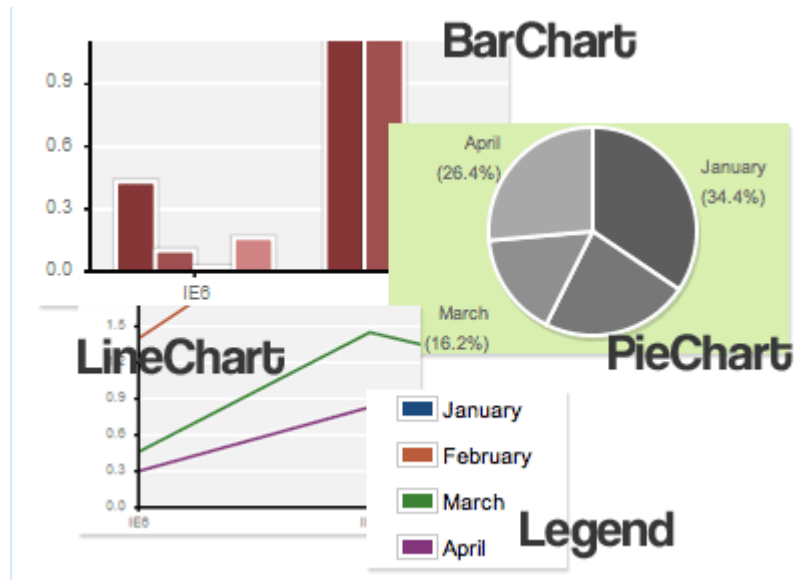


Obr.33: Ukážka PlotKit grafu

**Plotr** (<http://solutoire.com/plotr>)

**licencia:** BSD

Jednoduchý framework založený na PlotKite, ktorý vykresľuje základné grafy.



Obr.34: Ukážka PlotKit grafov

## 4. Návrh a realizácia riešenia

Táto časť je venovaná návrhu vlastného systému (ďalej len systém), ktorý by umožňoval vykresľovať grafy na internetových stránkach. Je rozdelená na 3 hlavné časti.

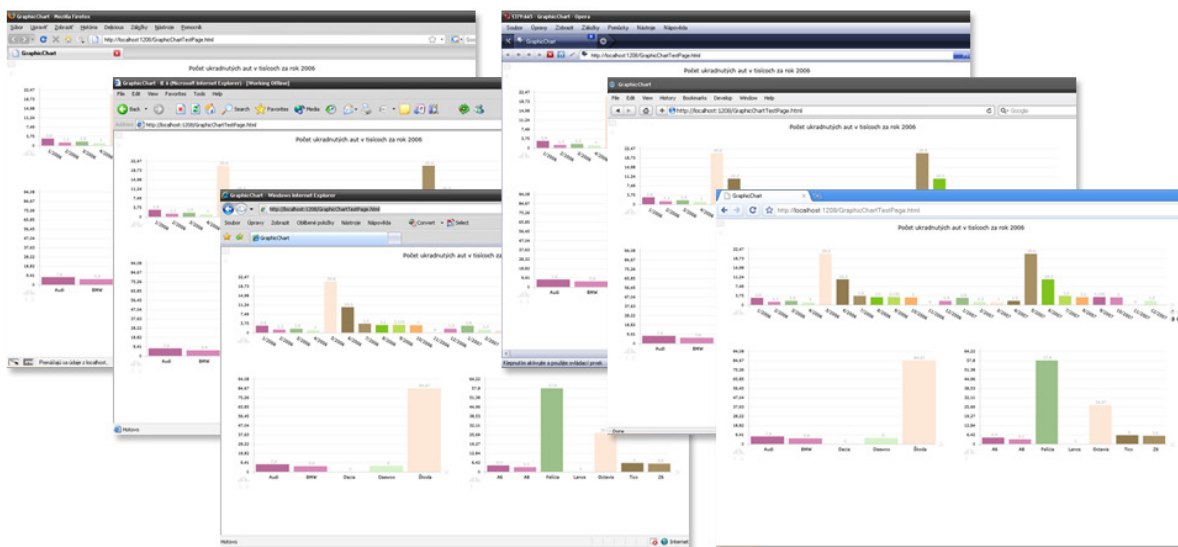
Prvá časť je venovaná špecifikácii. Je v nej podrobnejšie rozpísaná funkcionálna, najmä tá, ktorá sa líši od riešení, ktoré sú momentálne dostupné, prípadne je úplne nová.

Druhá časť je venovaná návrhu užívateľského prostredia [8], pričom sa zameriava na užívateľskú prívetivosť tohto systému a jednoduchosť používania. Popisuje podrobne taktiež význam grafických prvkov.

Posledná časť tejto kapitoly je venovaná samotnej implementácii systému a technikám, ktoré boli pri nej použité.

## 4.1. Špecifikácia

System bude napísaný v jazyku C# 3.0 s využitím technológie Silverlight 2.0, pričom musí byť korektné spustiteľný v každom prehliadači a operačnom systéme, ktorý ma podporu pre technológiu Silverlight 2.0, teda má korektné nainštalovaný potrebný plugin. Na obr.35 je vidieť ako Silverlight plugin zabezpečuje rovnaké zobrazenie obsahu bez ohľadu na typ prehliadača.



Obr.35: System spustený v prehliadačoch Mozilla Firefox 3, IE 6, IE 7, Opera 9.64, Safari 4 Beta, Google Chrome 2 na operačnom systéme Windows XP

### 4.1.1. Dátový zdroj

Ako dátový zdroj sa bude dať použiť súbor XML, prípadne komprimovaný súbor XML vo formáte ZIP. XML súbor, ako dátový zdroj, bol zvolený hlavne kvôli svojej rozšírenosti, ľahkej tvorbe, editácii a rýchlemu, intuitívnemu pochopeniu jeho štruktúry.

Štruktúra tohto súboru bude vychádza z vlastného návrhu, pričom bude rozdelená na dve časti. Prvá bude slúžiť na nastavenie parametrov jednotlivých grafov. Každý graf bude mať svoje nastavenia, nebudú žiadne spoločné. Naproti tomu v druhej dátovej časti budú štruktúrované vypísané dáta pre všetky grafy spoločne. Toto rozdelenie na časti má sprehľadniť dátovú štruktúru, kedy je dátová časť oddelená od časti určenej na nastavenie špecifických vlastností pre každý graf.

Pri tvorbe štruktúry XML súboru má byť braný ohľad hlavne na jej intuitívne pochopenie a minimalizovanie počtu potrebných záznamov.

#### 4.1.2. Vykresľovanie grafov

Systém bude vedieť zobrazovať stĺpcové 2D grafy, pričom musí byť pripravený na možnosť implementácie ďalších typov grafov. Systém musí vedieť zobrazit' súčasne viacero grafov, ktoré môžu byť vzájomne dátovo prepojené a reagovať na zmeny (v dátach) v ostatných grafoch. Maximálne sa budú dať zobrazit' 4 grafy súčasne, keďže vyšší počet grafov už považujeme za neefektívny.

#### 4.1.3. Prepojenia grafov

Prepojenie grafov bude spočívať v možnosti „zviazať“ grafy pomocou dát, ktoré majú spoločné. Znamená to, že ak užívateľ vyradí niektoré dáta na jednom grafe, zmena sa okamžite prejaví na všetkých ostatných. Vyradenie dát sa uskutoční kliknutím na príslušný stĺpec v grafe, pričom opätovné kliknutie bude mať za následok vrátenie dát do grafu.

Máme napríklad dva grafy, pričom jeden zobrazuje počet predaných áut po mesiacoch za celý rok a druhý typ áut, ktoré boli za ten rok predané. Vyradením napríklad mesiaca január zmení aj druhý graf svoj obsah a autá, ktoré boli predané v tomto mesiaci, nezobrazí. Takto sa budú dať prepojiť až 4 grafy.

#### 4.1.4. Zoskupovanie dát v grafoch

Ďalšia z funkcionalít, ktorá nie je v drvivej väčšine iných systémoch podporovaná, je zoskupovanie dát.

Každý graf v tomto systéme bude môcť pomocou ovládacích prvkov prepínať medzi alternatívnym obsahom X-ovej zložky grafu. Ak X-ová zložka bude obsahovať napríklad typy áut (Audi, Škoda, BMW,...) tak zapnutím alternatívneho obsahu sa môže zmeniť X-ová zložka napríklad na (Audi[kombi], Audi[sedan], Škoda[kombi], Škoda[sedan], BMW[kombi], BMW[sedan],...). Bude záležať od nastavenia grafu v XML súbore. Maximálne však bude mať každý graf k dispozícii dve takéto alternatívne osi, ktoré sa budú dať definovať v XML súbore. Tieto dve alternatívy budú môcť byť zapnuté aj spoločne.

Ďalším typom zoskupovania bude granularita. Ak bude nastavená X-ová os grafu ako DateTime, teda graf bude zobrazovať na tejto osi časové údaje, potom bude možné pomocou ovládacích prvkov zoskupiť dáta v takomto grafe po rokoch, polrokoch, kvartáloch, prípadne po mesiacoch. Pri zapnutej granularite bude minimálne a predvolené zobrazenie po mesiacoch.

#### 4.1.5. Zorad'ovanie

Aj keď by sa mohlo zdať, že zorad'ovanie by mal podporovať každý systém na vykresľovanie (stĺpcových) grafov, opak je pravdou. Táto, dalo by sa povedať základná funkcionálna nie je takmer vôbec podporovaná, pričom má veľké praktické využitie.

Nami navrhnutý systém bude podporovať zorad'ovanie údajov podľa Y-ovej osi vzostupne aj zostupne, s jednoduchou možnosťou návratu do pôvodného nastavenia. Systém bude v základnom zobrazení zorad'ovať graf podľa X-ovej osi, čo je napríklad veľmi žiaduce ak je X-ová os nastavená ako časová. Dáta môžu byť zapísané v rôznom chronologickom poradí a systém ich vždy automaticky utriedi.

#### 4.1.6. Funkcia Zoom a Full Screen

Do systému budú zakomponované aj tieto dve zaujímavé funkcionality, ktoré taktiež nie sú bežnou súčasťou momentálne dostupných riešení a príjemne zlepšujú prácu s grafmi.

Zapnutý Full Screen prepne grafy do celoobrazovkového módu, kedy sa nezobrazí ostatný obsah stránky, ako ani okno prehliadača, ale iba samotná Silverlight 2.0 aplikácia, ktorá vyplní celú plochu monitora.

Zoom, ako už názov napovedá dovoľí zväčšovať, približovať grafy, ako aj zmenšovať, oddaľovať. Táto funkcionálna, najmä v spojení so zapnutým režimom Full Screen výrazne zlepšuje použiteľnosť pre ľudí so zrakovým postihnutím.

Obidve tieto vlastnosti budú jednoducho použiteľné, cez na to určené tlačidlá.

#### 4.1.7. Save a Load

Toto je funkcionálna, ktorú nemá zakomponované žiadne z dostupných riešení. V tomto systéme bude implementovaný odľahčený variant Save a Load funkcionality, ktorý si dokáže uložiť (Save) prípadne znova načítať (Load) jednu sadu grafov. Teda pri príchode na stránku ak klikneme na „Save“ tlačidlo uložia sa nám grafy s aktuálnymi dátami. Ak prideme napríklad o týždeň, graf už môže pracovať s inými dátami. Po kliknutí na tlačidlo „Load“ sa nám však načítajú a vykreslia naposledy uložené grafy. Pri každom z týchto úkonov nás upozorní dialógové okno o úspešnosti uloženia, prípadne o úspešnosti nahradia grafov aj s dátumom ich uloženia.



#### 4.1.8. Scale Break

V časti 3.2. bolo predstavené riešenie na vykresľovanie grafov, ktoré umožňovalo zapnúť automatické Scale Breaks. Užívateľ si vedel zapnúť tieto Scale Breaks a tie sa podľa určitých nastavovacích premenných zobrazili. Interakcia s užívateľom bola však nulová.

Nami implementovaný systém bude používať postup, kedy sa pomocou tlačidla bude dať prejsť do „Scale Break” módu. Následne sa kliknutím na stípec určí miesto, kde by sa mal použiť Scale Break a systém vyhodnotí, či to je možné. Ak áno, vykreslí Scale Break. Následným opakovaným kliknutím sa Scale Break zruší.

#### 4.1.9. Implementácia do stránok

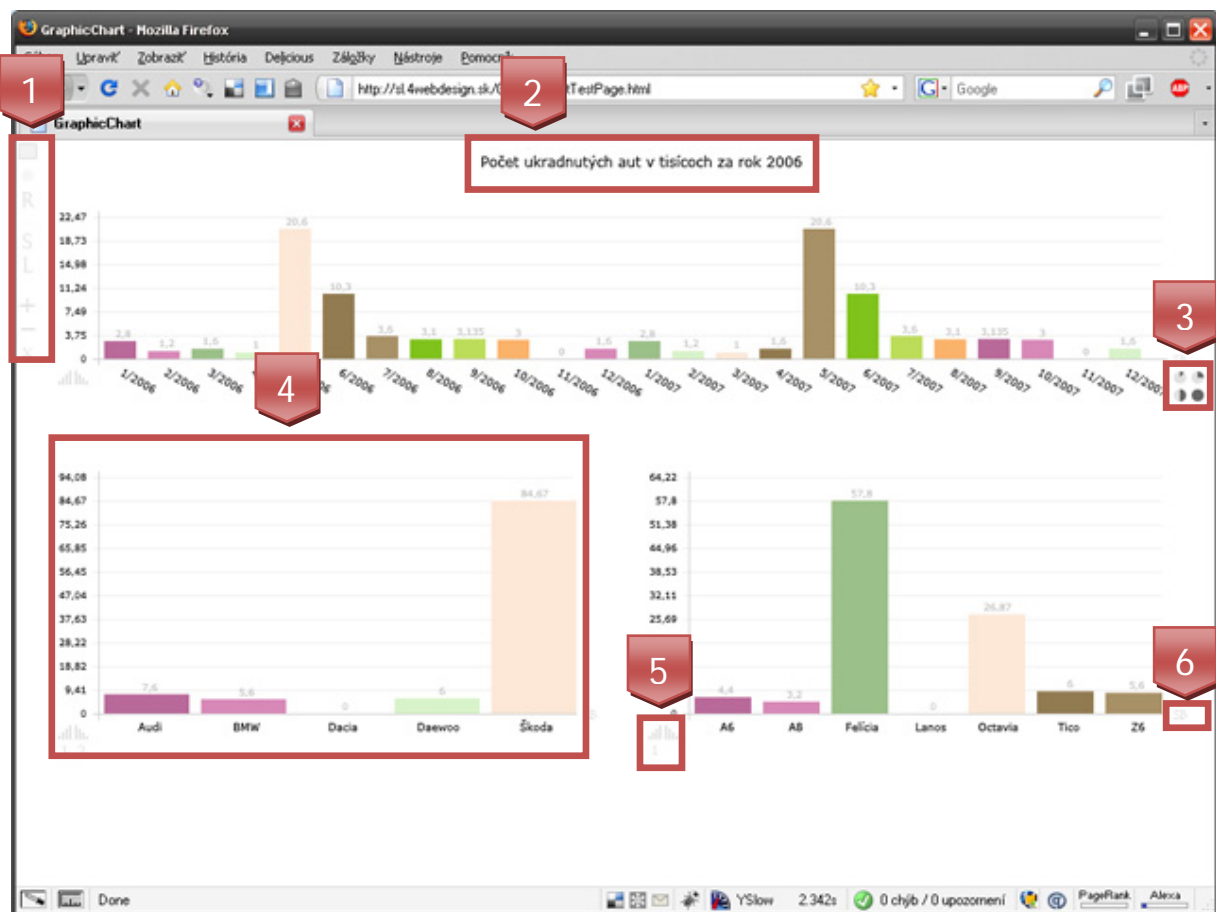
Implementácia do stránok bude prebiehať veľmi jednoducho. Na miesto v stránke, kde budú zobrazené grafy sa vloží kód s odkazom na .XAP súbor a odkaz na XML súbor s dátami a nastaveniami pre grafy.

Čo sa týka .XAP súboru, jedná sa o modifikovaný ZIP súbor, ktorý obsahuje všetky súbory XAML a DLL. Ide o štandardné knižnice, nie sú však vykonávané vo frameworku operačného systému, ale vo frameworku zásuvného modulu Silverlight. Pre overenie, stačí jednoducho zmeniť koncovku súboru z XAP na ZIP a okamžite sa dá overiť, čo tento archív obsahuje.

## 4.2. Návrh užívateľského prostredia

Všetky ovládacie prvky majú decentný vzhľad, aby zbytočne neodpúťavali od samotného obsahu grafu. Na obr.36 je vidieť HTML stránku, ktorá vykresľuje tri navzájom prepojené grafy. Užívateľské prostredie je rozdelené na nasledujúce časti:

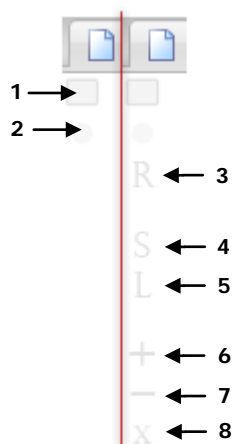
- 1 - Menu
- 2 - Titulok grafu
- 3 - Tlačidlá na zmenu granularity grafu
- 4 - Jeden z troch vykreslených grafov spolu s popiskami
- 5 - Tlačidlá pre zoradovanie a „alternatívne osi X“
- 6 - Tlačidlo na zapnutie a vypnutie Scale Break-u



Obr.36: Užívateľské prostredie systému

## 4.2.1. Menu

Hlavné menu sa nachádza v ľavej hornej časti systému a ponúka najdôležitejšie funkcie okamžite k dispozícii. Menu sa môže nachádzať v dvoch stavoch. Zobrazené a skryté. Na obr.37 je vidieť obidva



tieto stavy. Na obrázku naľavo od červenej čiary je vidieť menu v uzavretom stave. Vtedy sú vidieť iba dve tlačidlá. Všetky tlačidlá v menu majú nastavený ToolTip text (pomocný text), ktorý sa zobrazí pri prechode myšou cez tlačidlo.

Tlačidlo **č.1** slúži na prepnutie systému do Full Screen módu, teda prepne systém na celú obrazovku, kedy je zobrazená iba Silverlight aplikácia a nič iné.

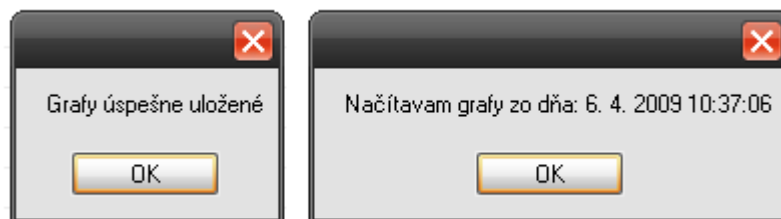
Tlačidlo **č.2** slúži na zobrazenie/skrytie zvyšnej časti menu. Po kliknutí na toto tlačidlo sa zobrazí menu do stavu, ako môžete vidieť na obrázku napravo od červenej čiary a následným ďalším kliknutím sa menu schová. Predvolene pri spustení systému je menu skryté.

Obr.37: Menu

Ak je celé menu zobrazené, ponúka rozšírenú sadu tlačidiel, ktoré uľahčujú prácu v tomto systéme.

Tlačidlo **č.3** vykoná „Reset“ grafu. Resetom je myslené obnovenie grafu do pôvodného stavu ako vyzeral pri načítaní stránky.

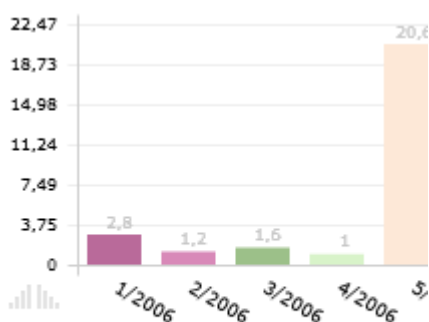
Nasleduje sada tlačidiel určených na „Load“ a „Save“. Tlačidlo **č.4** slúži na uloženie aktuálneho grafu. Naopak tlačidlo **č.5** načíta naposledy uložený graf. Na obr.38 je vidieť dialógové okná ktoré sa zobrazia pri uložení, respektíve načítaní grafov.



Obr.38: Dialógové okná pre Save a Load grafov

Pod týmito tlačidlami sa nachádza trojica tlačidiel určených na Zoom grafov. Tlačidlo **č.6** slúži na priblíženie grafov. Naopak tlačidlo **č.7** slúži na oddialenie grafov. Posledné tlačidlo **č.8** ma za následok vrátenie grafov do pôvodného stavu, po viacnásobnom použití tlačidiel „+“ a „-“.

## 4.2.2. Vykresľovanie grafov



Obr.39: Výsek z grafu

Na obr.39 je vidieť výsek z grafu v rovnakej mierke ako by bol viditeľný aj v prehliadači.

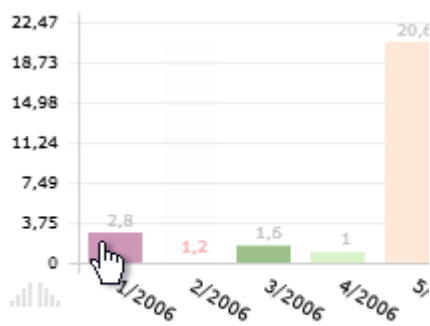
Popisok zobrazujúci hodnotu každého stĺpca sa nachádza nad stĺpcom a jeho farba je odtieň šedej. Odtieň je dostatočný, ale nebije príliš do očí.

Ako je vidieť z obrázku, popisky hlavnej X-ovej osi sú natočené pod uhlom. Ak je veľa dát, popisky sa automaticky prispôbia obsahu, prípadne môže nastať aj situácia, keď sa graf pokúša vykresliť príliš veľa dát. Šírka stĺpcov by vtedy musela klesnúť pod minimálnu hodnotu. Graf v takejto situácii nevykreslí obsah, ale oznámi pokus o zobrazenie príliš veľa dát.

Najvýraznejším „prvkom“ grafov sú samotné stĺpce. Na obr.40 je vidieť paleta farieb, ktorá je použitá pri ich zafarbovaní. Pozostáva z desiatich farieb, pričom boli vybrané hlavne pastelové farby. Keďže farby majú zväčša pastelové odtiene, neťahajú oči.



Obr.40: Paleta farieb



Obr.41: Výsek z grafu

Obr.41 v porovnaní s obr.39 zobrazuje rovnaký výsek grafu. Ako vidieť z obrázku, nastala zmena. Stĺpec zobrazujúci hodnotu pre „2/2006“ zmizol a nahradila ho iba hodnota 1,2, ktorá sa zobrazuje takmer na X-ovej osi. Tento stav nastal po kliknutí na tento stĺpec.

Takýto stĺpec je vyradený z dát, ktoré sa majú vykresliť a s ktorými pracujú aj ostatné grafy.

V tejto situácii stĺpec zmizne a je nahradený veľmi jemnou siluetou stĺpca a popiskom, ktorý má červený odtieň a je umiestnený až pri X-ovej osi. Tým je docielené dostatočne výrazne odlíšenie od normálnych stĺpcov, bez straty informácie o hodnote tohto stĺpca.

Ďalšou zmenou je stĺpec „1/2006“. Je na ňom kurzor myši a v porovnaní s rovnakým stĺpcom na obr.39 ma svetlejší odtieň farby.

Každý stĺpec má pri prechode myšou nastavený „Hover“ efekt. Teda ak sa prejde myšou cez plochu stĺpca farba zjemní svoj odtieň. Tento efekt spolu so zmenou kurzoru dáva intuitívne najavo „klikni na mňa , niečo sa stane“ a zlepšuje použiteľnosť systému.

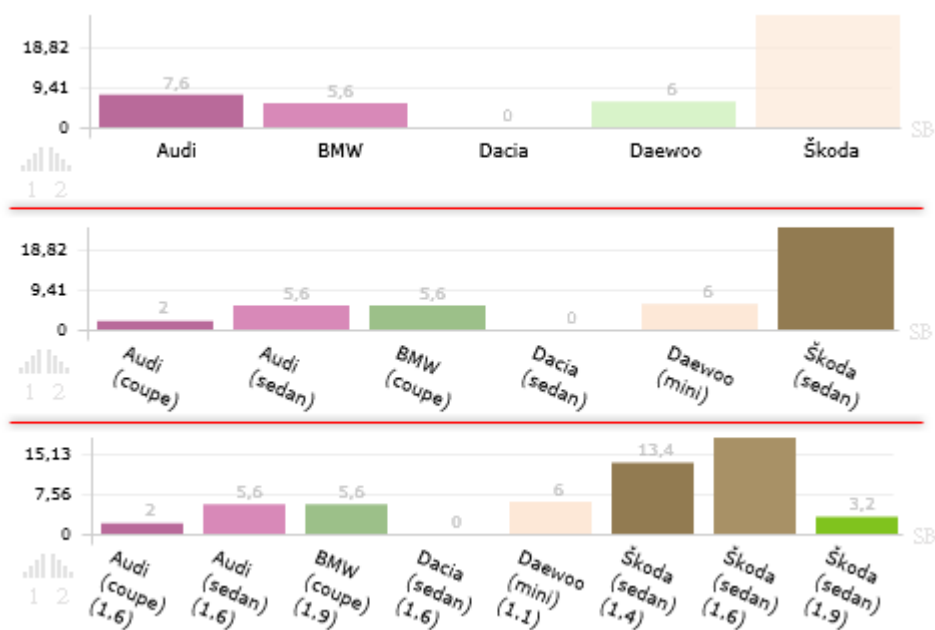
### 4.2.3. Zoskupovanie dát v grafoch

Obr.42 zobrazuje príklad alternatívnej osi X jedného grafu. V ľavej dolnej časti sa nachádzajú tlačidlá „1“ a „2“, ktoré umožňujú prepínanie medzi alternatívnymi osami grafov. Presnejšie ide o akési ďalšie zjemnenie X-ovej zložky grafu.

V XML súbore má každý graf nastavené číslo stĺpca, ktorý slúži ako dátový zdroj pre X-ovú os a taktiež číslo stĺpca, ktorý slúži ako zdroj dát pre Y-ovú os. V XML súbore sú však ešte dva ďalšie tagy, do ktorých sa píše číslo stĺpca, ktorý ma slúžiť ako zdroj pre alternatívnu os X.

Ako vidieť na príklade, každú značku auta (Audi, BMW,...) vieme ďalej ešte rozdeliť do podrobnejších kategórií, ako napríklad typ karosérie (coupe, sedan,...) alebo objem motora (1.6, 1.9,...).

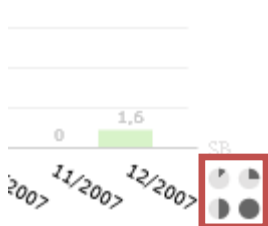
Prvá časť obrázku ukazuje graf v klasickom zobrazení. Druhá časť ukazuje, ako vyzerá graf po kliknutí na „1“, keď sa značky ďalej rozdelili podľa typu karosérie. Nakoniec v poslednej časti je vidieť graf po tom, čo bolo kliknuté najprv na „1“ a následne na „2“.



Obr.42: Alternatívne grafy

Ako teda vidieť z obrázku, môžu byť zapnuté samostatne „1“, alebo „2“, prípadne obidve tieto alternatívy súčasne. Keďže môžu byť na stránke maximálne 4 grafy a každý môže mať celkovo až 4

rôzne zobrazenia (klasické, zapnuté „1“, zapnuté „2“, zapnuté obidve súčasne), tak dostaneme až 16 rôznych grafov, čo už postačuje aj na zložité a komplexné štatistiky.

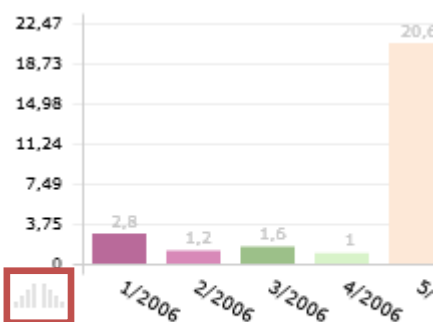


Obr.43: Granularita

Ďalšou možnosťou zoskupovania dát v grafe je granularita. Obr.43 ukazuje v červenom rámečku 4 ovládacie prvky na zmenu granularity. Tieto tlačidlá sa automaticky zobrazia, ak je X-ová os nastavená ako časová (DateTime). Nachádzajú sa v pravej dolnej časti grafu.

Ikonky majú veľmi intuitívny a príjemný vzhľad. Prvá ikonka (vľavo hore) prepne do zobrazenia po mesiacoch, ktoré je automaticky nastavené ako predvolené zobrazovanie. Ďalšia ikonka (vpravo hore) prepne do zobrazenia po kvartáloch. Tretia (vľavo dole) prepne na polroky a posledná prepne graf na zobrazenie po rokoch.

#### 4.2.4. Zorad'ovanie



Obr.44: Zorad'ovanie

Na obr.44 je vidieť v červenom rámečku dvojica ovládacích prvkov slúžiacich na zoradenie grafu podľa Y-ovej osi. Prvý ovládací prvok slúži na vzostupné zoradenie. Druhý zase na zostupné zoradenie. Opakovaným stlačením tlačidla na zorad'ovanie, sa vrátíme do pôvodného stavu.

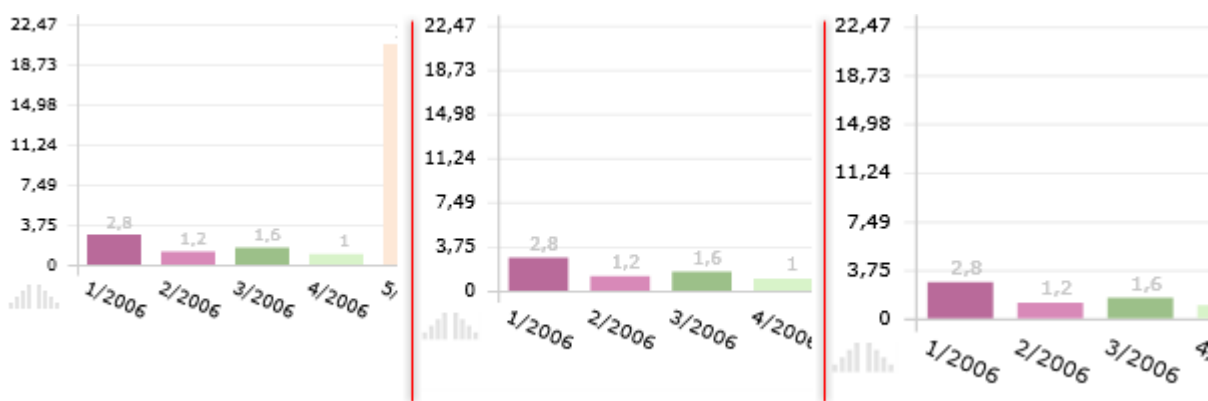
Design týchto ovládacích prvkov vychádza zo všeobecne zaužívaných konvencií a je intuitívne pochopiteľný.

#### 4.2.5. Funkcia ZOOM a Full Screen

V časti 4.2.1. bol popísaný význam funkcie Full Screen. Taktiež tam bol popísaný ovládací prvok slúžiaci na prechod do Full Screen módu.

V časti 4.2.1. boli tiež predstavené ovládacie prvky, ktoré umožňujú funkciu ZOOM. Boli to tri tlačidlá „+“, „-“ a „x“. Prvé dve intuitívne zväčšujú, alebo znižujú graf. Posledné slúži ako taký reset ZOOMu a návrat do pôvodnej mierky.

Obr.45 demonštruje použitie funkcie ZOOM. Prvý graf je zobrazený v klasickej mierke, prípadne po použití tlačidla „x“. Druhý graf sa nachádza v stave, keď bolo 2x stlačené tlačidlo „+“ a nakoniec tretí graf ukazuje ako vyzerá graf po štyroch stlačeniach tlačidla „+“.



Obr.45: Použitie funkcie ZOOM

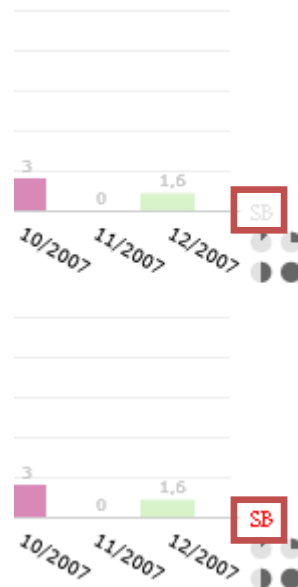
Ako vidieť z obrázka, v prípade použitia funkcie ZOOM nenastane extrémne veľká zmena v mierke grafu. Koeficient, ktorý určuje pomer zväčšenia (zmenšenia) je nastavený tak, aby si každý vedel nastaviť sebe vyhovujúce zobrazenie. Či už ide o malé zväčšenie, alebo o niekoľkonásobné použitie „+“ pre ľudí so zrakovou vadou. Jediným obmedzujúcim faktorom je veľkosť plochy na stránke určenej pre Silverlight aplikáciu, prípadne iba veľkosť monitora, ak je systém prepnutý do Full Screen módu. Pri veľkom zväčšení sa graf nemusí vojsť do priestoru na zobrazenie, preto treba túto funkčnosť vhodne používať.

#### 4.2.6. ScaleBreak

Tlačidlo, ktoré slúži na zapnutie ScaleBreak sa nachádza v pravej dolnej časti grafu, ako je vidieť na obr.46 v červenom rámečku. Ak je zapnuté, jeho farba sa zmení na červenú.

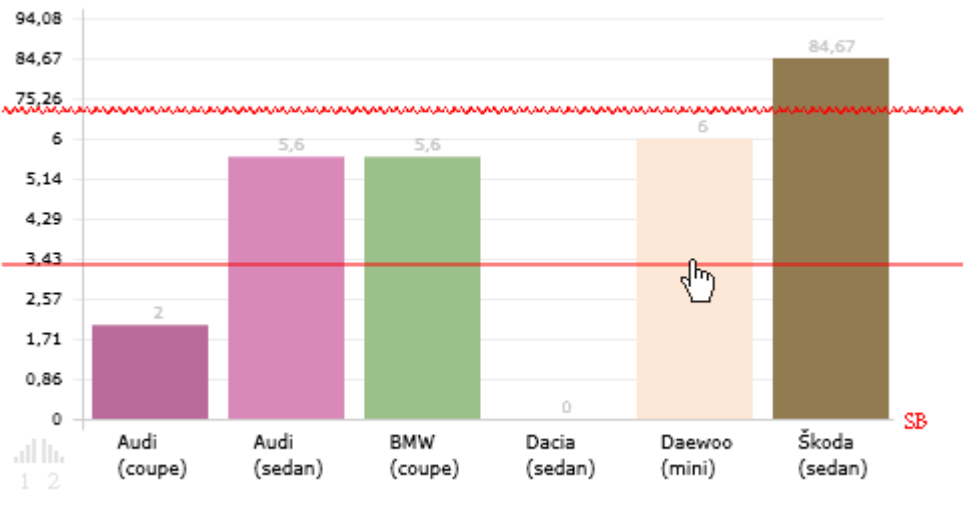
Zapnutím ScaleBreak sa vypne možnosť kliknutím na stĺpec grafu selektovať hodnotu a celý graf sa prepne do režimu ScaleBreak, kedy sa pri prechode myšou cez graf zobrazí pomocná červená čiara, rovnobežná s X-ovou osou a prechádzajúca cez všetky stĺpce grafu.

Táto pomocná čiara slúži na lepšie použitie ScaleBreak, keďže sa vizuálne dá aj v širšom grafe overiť, ktoré hodnoty by boli nad ScaleBreak a ktoré pod ScaleBreak čiarou.



Obr.46: ScaleBreak tlačidlo

Následným kliknutím na niektorý zo stĺpcov sa overí, či je možné na ňom vykresliť ScaleBreak. Ak áno, tak sa zobrazí ScaleBreak čiara, upraví sa popis Y-ovej osi a graf sa prekreslí do novej podoby. Následným opakovaným kliknutím na akýkoľvek stĺpec sa ScaleBreak zruší. Obr.47 ukazuje použitie ScaleBreak.



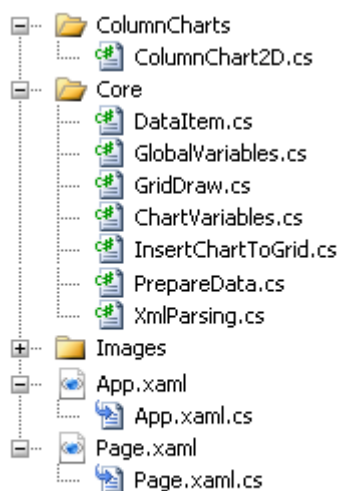
Obr.47: Použitie ScaleBreak



## 4.3. Implementácia

Pri vývoji tohto systému bol použitý programovací jazyk C# 3.0 [7, 13] a technológia Silverlight 2.0 [1, 2, 3, 5, 11]. Užívateľské prostredie bolo z časti vytvorené pomocou XAML kódu, ale z väčšej časti je generované priamo za behu pomocou C# kódu.

### 4.3.1. Fyzická štruktúra



Obr.48: Štruktúra projektu

Obr.48 ukazuje adresárovú štruktúru projektu. Zložka ColumnCharts obsahuje zdrojové kódy na vykresľovanie špecifických typov grafov. ColumnCharts2D.cs obsahuje zdrojový kód špecifický pre vykresľovanie stĺpcových 2D grafov.

Zložka Core obsahuje najdôležitejšie triedy potrebné na správne vykreslenie grafov. Od parsovania XML súborov až po samotné nešpecifické vykresľovanie grafov na obrazovku.

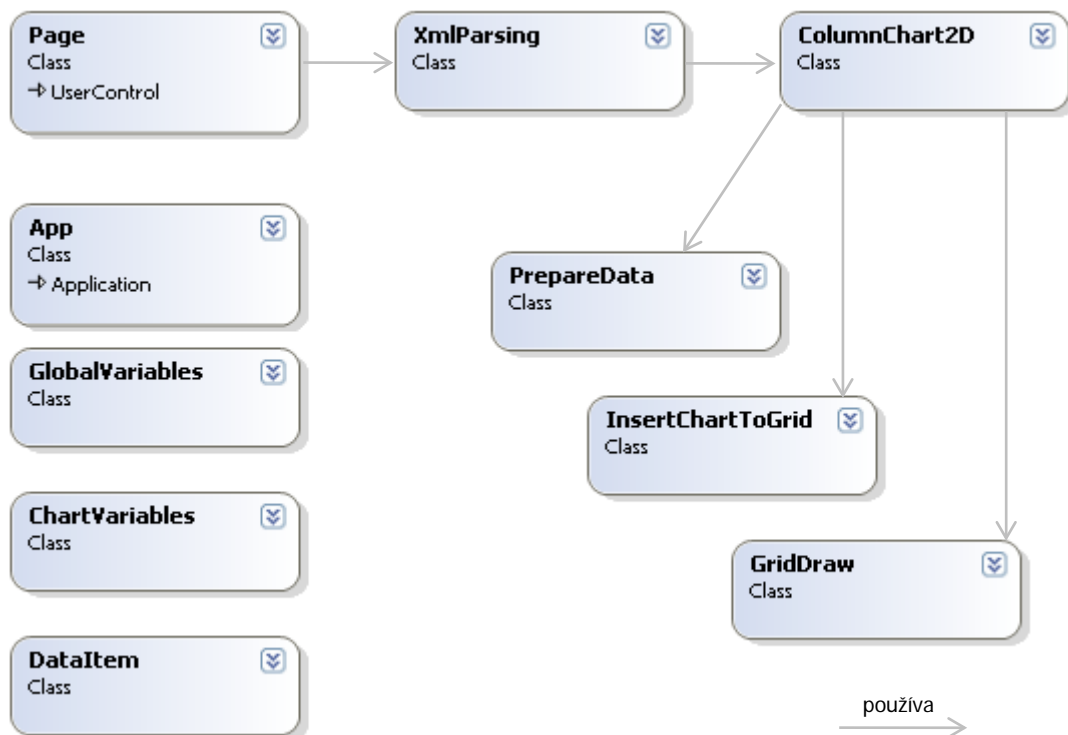
V zložke Images sa nachádzajú všetky obrázky, ktoré sú potrebné na správne zobrazenie systému. Jedná sa najmä o pozadia pre rôzne tlačidlá, ktoré sa zobrazujú pri grafoch a v menu systému.

Súbor Page.xaml.cs obsahuje počiatočné grafické rozhranie, ktoré sa načíta, keď sa aplikácia spustí.

App.xaml vo všeobecnosti slúži na deklarovanie zdrojov (Resources), ako sú rôzne štetce, šablóny vzhľadov, obrázky,... Nachádzajú sa tu zdroje, ktoré sa dajú použiť v celej aplikácii, na rôznych miestach viackrát. V súbore App.xaml.cs sa zas dajú odchytiť rôzne udalosti na úrovni aplikácie, ako napríklad Application\_Startup (pri načítaní aplikácie), Application\_Exit (pri ukončení aplikácie). Nami navrhnutý systém využíva App.xaml na uchovávanie šablón pre tlačidlá.

### 4.3.2. Popis súborov a tried

Všetky triedy, metódy ako aj premenné majú komentáre. Taktiež niektoré časti kódu, ktoré by mohli byť nejasné obsahujú komentár. Obr.49 pre lepšie pochopenie vzťahov medzi triedami ukazuje ich vzájomnú prepojenosť.



Obr.49: Jednoduchý Class diagram systému

- DataItem.cs** - Pomocná trieda. Je využitá pri zoradovaní prvkov ako pomocná štruktúra na uchovávanie dát určených na zoradenie.
- ChartVariables.cs** - V tejto triede sa nachádzajú premenné, ktoré sú špecifické pre každý graf. Sú na inicializované pomocou hodnôt z XML súboru. Ide napríklad o veľkosť grafu a pod. Nachádza sa tu však aj niekoľko hodnôt, ktorým sa nepriraduje hodnota z XML súboru, ale hodnotu už majú inicializovanú pri vytvorení, ako napríklad predvolené nastavenie granularity grafu na mesiace.
- GlobalVariables.cs** - Trieda obsahujúca globálne premenné, ako napríklad farby stĺpcov v grafe, maximálnu šírku stĺpca, chybové hlášky,... Jedná sa o premenné spoločné pre všetky grafy.
- Page.xaml.cs** - Obsahuje počiatkové grafické rozhranie, ktoré sa načíta keď sa aplikácia spustí. Taktiež obsahuje celé hlavné menu aplikácie aj s jeho funkcionalitou.
- XmlParsing.cs** - Trieda slúžiaca na parsovanie XML súboru s dátami pre graf. Trieda obsahuje metódy na spracovanie súboru, jeho rozparsovanie, spracovanie dát a následne podľa toho, aký typ grafu sa má vykresliť, vytvorí inštanciu potrebnej triedy (napr. ColumnChart2D).

- ColumnChart2D.cs** - Obslužná trieda, ktorá zabezpečuje vykreslenie 2D stĺpcového grafu. Používa metódy tried PrepareData, InsertChartToGrid a GridDraw.
- PrepareData.cs** - Obsahuje metódy na spracovanie dátovej zložky grafu, pričom tu sa berú v úvahu rôzne selekcie, zoradovanie. Obsahuje dve metódy, ktoré vracajú dáta pre X-ovú os a Y-ovú os.
- InsertChartToGrid.cs** - Trieda, ktorá podľa nastavenia grafu rozhoduje kam ho na ploche umiestniť.
- GridDraw.cs** - Najdôležitejšia trieda, ktorá sa stará o samotné vykreslenie grafu na stránky. Obsahuje množstvo metód, ktoré sa starajú o samotné vykresľovanie grafu. Taktiež obsahuje rôzne pomocné metódy, ktoré sa starajú o výpočty a prípravu dát pre metódy vykresľujúce graf. Napríklad sú tu metódy určené na vykresľovanie súradnicových sietí, vykresľovanie popiskov, vyrátanie, či je možné použiť ScaleBreak....

### 4.3.3. Dátový zdroj

Ako dátový zdroj, ako už bolo spomenuté, môže byť použitý XML súbor, prípadne komprimovaný XML súbor vo formáte ZIP. Na parsovanie súboru je použitý LINQ, ktorý je popísaný v ďalšej časti.

Pri väčších súboroch XML je vhodnejšie použiť komprimovaný formát, keďže sa tým rapídne zmenší výsledný súbor a tým aj zmenší objem dát prenášaných k užívateľovi na PC. ZIP súbor musí mať názov rovnaký ako XML súbor, ktorý je v ňom. Napríklad „test.zip“ musí obsahovať súbor „test.xml“.

Na obr.50 je ukážka jednoduchého XML súboru, ktorý obsahuje definíciu jedného grafu a iba jednu položku v tomto grafe.

Štruktúra začína klasickou definíciou XML súboru:

```
<?xml version="1.0" encoding="utf-8" ?>
```

Nasleduje tag <root>, ktorý v sebe obsahuje tagy <settings> a <datasection>.

Tag settings obaluje definície grafov. V tagu datasection sú zasa spoločné dáta na zobrazenie pre všetky grafy.

```

<?xml version="1.0" encoding="utf-8" ?>
<root>
  <settings>
    <chart>
      <id>1</id>
      <charttype>ColumnChart2D</charttype>
      <xdata>3</xdata>
      <ydata>4</ydata>
      <alternativechartfirst>0</alternativechartfirst>
      <alternativechartsecond>0</alternativechartsecond>
      <isxdatadatetetime>true</isxdatadatetetime>
      <datetimesource>3</datetimesource>
      <positionrow>top</positionrow>
      <posiitoncolumn>center</posiitoncolumn>
      <tittle>Počet ukradnutých aut v tisícoch za rok 2006</tittle>
      <showtittle>true</showtittle>
      <width>1000</width>
      <height>230</height>
    </chart>
  </settings>
  <datasection>
    <row>
      <column>Audi</column>
      <column>A6</column>
      <column>1/2/2006</column>
      <column>1,2</column>
      <column>sedan</column>
      <column>1.6</column>
    </row>
  </datasection>
</root>

```

Obr.50: Ukážka štruktúry XML súboru

#### 4.3.3.1. Tag <settings>

Tag settings obsahuje definície samotných grafov. Každý graf musí mať korešpondujúci tag <chart> v tejto sekcii a korektné vypísané **všetky** doleuvedené tagy, slúžiace na korektné nastavenie a vykreslenie grafu.

Nižšie je podrobne rozpísaný význam každého tagu, spolu s možnými hodnotami, ktoré mu môžu byť priradené.

- |                 |   |
|-----------------|---|
| <b>id</b>       | - Integer, číslo, ktoré jednoznačne identifikuje konkrétny graf.<br>Príklad: 1,2,3,...                    |
| <b>chartype</b> | - Určuje typ grafu, ktorý sa má vykresliť.<br>Príklad: <i>ColumnChart2D</i>                               |
| <b>xdata</b>    | - Určuje číslo stĺpca, ktorý je použitý ako dátový zdroj pre X-ovú os.<br>Číslovanie stĺpcov začína od 1. |

- Príklad: *1,2,3,...*
- ydata** - Určuje číslo stĺpca, ktorý je použitý ako dátový zdroj pre Y-ovú os. Číslovanie stĺpcov začína od 1 a hodnoty v stĺpci musia byť konvertovateľné na typ double.
- Príklad: *1,2,3,...*
- alternativechartfirst** - Určuje, či sa má povoliť zobrazenie alternatívnych grafov. Ak je nastavená 0, tak je možnosť zobrazenia alternatívneho grafu na prvej pozícii vypnutá. Iné prirodzené číslo, väčšie ako 0 určuje stĺpec, z ktorého sa majú brať dáta pre X-ovú osu.
- Príklad: *0,1,2,3,...*
- alternativechartsecond** - Určuje, či sa má povoliť zobrazenie alternatívnych grafov. Ak je nastavená 0, tak je možnosť zobrazenia alternatívneho grafu na druhej pozícii vypnutá. Iné prirodzené číslo, väčšie ako 0 určuje stĺpec, z ktorého sa majú brať dáta pre X-ovú osu.
- Príklad: *0,1,2,3,...*
- isxdatadatetime** - Nastavenie tejto hodnoty na true, má za následok zapnutie zobrazovania granularity v celom systéme. V tomto stave sa hodnoty zo stĺpca z premennej datetimesource považujú za dátum (datetime). Ak následne v niektorom z grafov sa hodnota xdata rovná hodnote datetimesource, tak tento graf zobrazí granularitu a X-ová os bude následne považovaná za časovú. Z toho vyplýva nutnosť zadávania hodnôt do príslušného stĺpca, ktorý je braný ako zdroj údajov pre X-ovú os v tvare, ktorý je konvertovateľný na datetime (napr. v tvare 1/12/2006). V systéme môže byť iba jeden stĺpec typu datetime a každý graf musí mať tento tag, spolu s tagom datetimesource nastavený na rovnakú hodnotu.
- Príklad: *true, false*
- datetimesource** - Určuje číslo stĺpca, ktorý je braný ako zdroj pre časovú X-ovú os. Ak je nastavené na 0, je zobrazovanie X-ovej osi ako časovej vypnuté. Túto hodnotu musia mať všetky grafy nastavenú kvôli implementácii.
- Príklad: *0,1,2,3,...*

**positionrow** - Určuje riadok, v ktorom sa má graf vykresliť.

Príklad: *top, down*

top left	top right
down left	down right

**positioncolumn** - Určuje bunku v riadku, v ktorej sa má graf vykresliť. Ak je vybraná hodnota center, tak sa bunky left a right zlúčia do jednej a v tejto zjednotenej bunke sa zobrazí graf.

Príklad: *left, right, center*

top center
down center

Obr.51: positionrow a positioncolumn

**tittle** - Tag obsahujúci textový titulok, ktorý sa môže zobrazíť nad grafom v strede. Záleží od nastavenia tagu showtittle, či sa bude tento popisok zobrazovať, alebo nie.

Príklad: *string aj s diakritikou v kódovaní UTF-8*

**showtittle** - Nastavuje zapnutie/vypnutie zobrazovanie tittle v grafe. Každý graf môže mať vlastný titulok. Keď sa na stránke nachádza viacero grafov, je niekedy vhodné zobrazíť titulok iba u niektorých.

Príklad: *true, false*

**width** - Určuje šírku grafu v pixloch. Je dôležité brať do úvahy, že každý graf potrebuje približne 100 pixlov na miesto pre popisky a pomocný voľný priestor. Ďalej treba brať do úvahy množstvo zobrazených informácií. Pokiaľ by bolo informácií príliš veľa, graf sa nevykreslí a zobrazí chybovú hlášku.

Príklad: *200,450,...*

**height** - Určuje výšku grafu v pixloch. Je dôležité brať do úvahy, že každý graf potrebuje približne 100 pixlov na miesto pre popisky a pomocný voľný priestor.

Príklad: *150,200,...*

#### 4.3.3.2. Tag <datasection>

Tag `datasection` obsahuje definície dát, ktoré slúžia na napĺňanie grafov. Štruktúra je podobná ako štruktúra tabuľky v HTML, kde tag `datasection` je ekvivalentný s tagom `table`. Tag `row` je ekvivalentný s tagom `tr` a tag `column` je ekvivalent k tagu `td`.

Dôležité je dodržiavať postupnosť dát, aby v každom stĺpci boli vždy rovnaké dáta. Taktiež musí byť v každom riadku rovnaký počet buniek.

Takáto štruktúra bola vybraná kvôli jednoduchosti pochopenia a aj kvôli jednoduchosti generovania, napríklad aj z externej aplikácie.

```
<row>
  <column>Audi</column>
  <column>A6</column>
  <column>1/2/2006</column>
  <column>1,2</column>
  <column>sedan</column>
  <column>1.6</column>
</row>
<row>
  <column>BMW</column>
  <column>Z6</column>
  <column>1/1/2006</column>
  <column>2,8</column>
  <column>coupe</column>
  <column>1.9</column>
</row>
```

Obr.52: Obsah tagu `datasection`

#### 4.3.3.3. LINQ

V .NET aplikáciách je možné s relačnými, či hierarchickými dátami pracovať niekoľkými spôsobmi. V prípade dát, ktoré sú uložené v relačnej databáze je možné pristupovať k nim cez ADO.NET ako v pripojenom režime, tak aj v odpojenom.

Pokiaľ sú dáta reprezentované hierarchicky vo formáte XML, je možné využiť assembly *System.Xml* a pracovať s týmito dátami pomocou DOM, alebo ich spracovávať sekvenčne.

LINQ (Language INtegrated Query) [15, 16] je integrovaný jazyk, ktorý prináša nový spôsob práce s dátami na platforme .NET. LINQ prináša podporu dotazovania nad akýmikoľvek dátami, uľahčuje ich tvorbu, triedenie, prepojovanie a vyhľadávanie v nich.

Pomocou tohto nového prístupu je možné pracovať takmer s akýmikoľvek dátami, lebo architektúra LINQ je navrhnutá tak, že je možné tvoriť jej implementácie pre jednotlivé dátové zdroje. Je to podobné ako v ADO.NET, kde je možné implementáciou rozhrania vytvoriť .NET providera pre špecifický typ databázy. Pri technológii LINQ však táto rozšíriteľnosť presiahla relačné databázy a stala sa omnoho abstraktnejšia. Táto voľnosť je dosiahnutá hlavne vďaka novinke Expression Tree v .NET 3.5, ktorá umožňuje pracovať s kódom ako s dátami. LINQ zjednocuje prácu nad dátami rôzneho druhu.

Spolu s .NET frameworkom 3.5 je dodávaných niekoľko oficiálnych implementácií LINQ. Tieto implementácie by mali pokryť základné potreby pri tvorbe aplikácií. Jedná sa o:

- **LINQ to Objects** – implementácia LINQ pre štandardné kolekcie napr. nachádzajúce sa v pamäti
- **LINQ to SQL** – Implementácia LINQ pre Microsoft SQL Server 2000 a vyšší
- **LINQ to XML** – Implementácia LINQ pre prácu s XML dátami
- **LINQ to DataSet** – Implementácia LINQ pre prácu s ADO.NET datasetmi

Obr.53 ukazuje jednoduché použitie LINQ. Z polia *slova* sa vyberú všetky reťazce, ktorých dĺžka je menšia ako 5 a taktiež toto pole je zoradené podľa dĺžky slov.

```
string[] slova = { "motorka", "auto", "lietadlo", "ponorka", "kolobežka" };
var kratkeSlova = from c in slova
                  where c.Length < 5
                  orderby c.Length
                  select c;

foreach (string slovo in kratkeSlova)
{
    Console.WriteLine(slovo);
}
```

Obr.53: Ukážka použitia LINQ

V navrhnutom systéme je použitý LINQ to XML pre prístup do dátového XML súboru. Obr.54 zobrazuje použitie LINQ to XML na spracovanie nastavovacích premenných pre každý jeden graf.

```
var es = from b
in ThisDoc.Descendants("chart")
select new ChartVariables
{
    ID = Convert.ToInt32(b.Element("id").Value),
    ChartType = b.Element("charttype").Value,
    XData = (Convert.ToInt32(b.Element("xdata").Value) - 1).ToString(),
    YData = (Convert.ToInt32(b.Element("ydata").Value) - 1).ToString(),
    AlternativeChartFirst =
    (Convert.ToInt32(b.Element("alternativechartfirst").Value) - 1).ToString(),
    AlternativeChartSecond =
    (Convert.ToInt32(b.Element("alternativechartsecond").Value) - 1).ToString(),
    IsXDataDateTime = Convert.ToBoolean(b.Element("isxdatadatetime").Value),
    DateTimeSource =
    (Convert.ToInt32(b.Element("datetimesource").Value) - 1).ToString(),
    PositionRow = b.Element("positionrow").Value,
    PositionColumn = b.Element("posiitoncolumn").Value,
    ChartHeading = b.Element("tittle").Value,
    ShowDescription = Convert.ToBoolean(b.Element("showtittle").Value),
    ChartWidth = Convert.ToInt32(b.Element("width").Value),
    ChartHeight = Convert.ToInt32(b.Element("height").Value),
};
```

Obr.54: LINQ to XML použitý v systéme



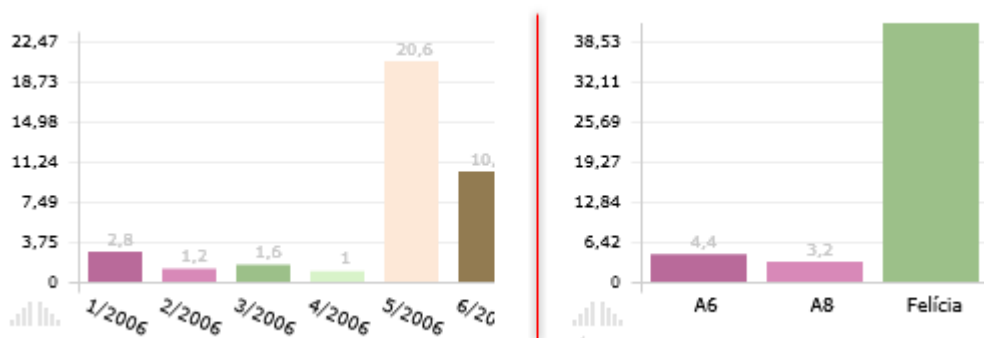
#### 4.3.4. Vykresľovanie grafov

Na prezentovanie obsahu Silverlight aplikácií slúžia tri typy Layout kontajnerov. Každý je určený na iný účel a spolu tvoria komplexnú sadu kontajnerov. Jedná sa o *StackPanel*, *Grid* a *Canvas*.

- StackPanel** - Zobrazuje elementy v horizontálnom, alebo vertikálnom zásobníku. Tento kontajner je väčšinou používaný pre malé celky, ktoré sú súčasťou väčšieho a komplexnejšieho systému.
- Grid** - Usporiadaná elementy do riadkov a buniek podľa neviditeľnej tabuľky. Je to jeden z najflexibilnejších a najpoužívanějších kontajnerov v Silverlighte.
- Canvas** - Umožňuje vkladať elementy na absolútnu pozíciu použitím fixných koordinátov. Je to najjednoduchší element, avšak s najmenšou flexibilitou.

Tu predstavený systém využíva všetky typy kontajnerov. Na hlavné menu je použitý vertikálny StackPanel. Každý graf sa presne vykresľuje do Canvas-u a následne je umiestnený do Grid-u, ktorý slúži na rozmiestnenie celých grafov.

Na samotné vykreslenie stĺpcov grafu sa používa Silverlight-ovský prvok Rectangle a na všetky čiary sa používa prvok Line. Na vykreslenie všetkých popiskov grafov je použitý TextBlock. Popisky X-ovej osi sa dynamicky prispôbujú obsahu. Ak sú stĺpce príliš úzke a popisky by sa prekryvali, tak sa následne všetky popisky zobrazia pod uhlom 24 stupňov, čo zabezpečí ich čitateľnosť aj pri užších stĺpcoch. Uhol 24 stupňov je postačujúci uhol na to, aby sa popisky neprekrývali ani pri veľmi úzkych stĺpcoch a na druhej strane, dostačujúci na to, aby sa zachovala dobrá čitateľnosť popiskov aj pri dlhších názvoch. Obr.55 ukazuje rozdiel medzi normálnym zobrazením popiskov a pootočeným. Dôležité je mať na pamäti pri tvorbe názvov popiskov fakt, že miesto na zobrazenie popiskov je ohraničené, nie je neobmedzene veľké. Preto treba voliť kratšie, výstižnejšie názvy, aby sa vošli do miesta určeného pre grafy.



Obr.55: Pootočenie popiskov

### 4.3.5. Prepojenia grafov

Pri parsovaní XML súboru sa pre každý graf vytvorí trieda `ChartVariables`. Taktiež sa vytvorí spoločná konštrukcia pre všetky grafy `List<List<string>>`, ktorá slúži na uchovávanie dátovej časti všetkých grafov. Simuluje vlastne tabuľku, kde vnútorný `List` sú bunky a vonkajší tvorí riadky. `List` ako dátová štruktúra bol vybraný hlavne kvôli jeho ľahkej použiteľnosti. Keďže grafy nebudú pracovať s extrémne veľkými dátami, nemuseli sme pri výbere dátovej štruktúry zohľadňovať rýchlosť.

Všetky grafy pracujú s takouto dátovou štruktúrou, pričom nad ňou robia selekciu dát podľa ďalších premenných, ktoré sú posielané spolu s touto štruktúrou. Napríklad zoznam vyradených hodnôt, čo sú hodnoty, na ktoré bolo kliknuté.

Každý graf si nad svojimi dátami vykoná všetky potrebné selekcie a podľa toho zobrazí graf. Ak je kliknuté na určitý stĺpec v grafe, tento stĺpec sa pripojí k zoznamu vybraných stĺpcov. Každý graf sa následne prekreslí s tým, že zoberie do úvahy nový zoznam vybraných hodnôt.

Samotné načítanie XML súboru prebieha v systéme iba raz. V opačnom prípade sa stále pracuje s kópiu dát, ktoré boli prvýkrát načítané z XML súboru. Tým je markantne zvýšená rýchlosť systému a znížená potreba prenášania dát zo serveru.

### 4.3.6. Zoskupovanie dát

Zoskupovanie dát ako granularita a alternatívne grafy spočíva v úprave kópie dátovej štruktúry `List<List<string>>`. Zoskupovanie spočíva v zmene tejto štruktúry, keď sa hodnoty, ktoré majú byť v spoločnej skupine prepíšu „názvom tejto skupiny“. Vždy sa prepisujú údaje iba v kópii tejto štruktúry, s ktorou sa ďalej pracuje a originál je uchovaný, aby sa z neho prípadne mohla urobiť znova kópia.

Napríklad pri zobrazení grafu po kvartáloch sa každá z hodnôt *1/1/2009*, *1/2/2009*, *1/3/2009* v tejto štruktúre prepíše hodnotou *2009 (1. kvart.)*. Následným ďalším spracovaním tejto štruktúry sa hodnoty stĺpcov s rovnakým názvom sčítajú a dostaneme iba jednu hodnotu pre tento kvartál.

### 4.3.7. Zoradovanie

Informáciu o zapnutom zoradovaní v sebe nesie `String`-ová premenná *sort*. Rovnako efektívne by sa dal miesto tejto premennej použiť `Enum`, my sme sa však rozhodli použiť túto premennú, ktorá môže byť prázdna, vtedy je graf zoradený podľa X-ovej osi. Môže byť nastavená na „d“ (descending), čo

znamená zostupné zoradenie podľa Y-ových hodnôt grafu. Prípadne môže byť nastavená na „a“ (ascending), čo znamená vzostupné zoradenie.

Nastavenie tejto premennej sa zohľadňuje pri triedení grafu, kde sa berú do úvahy vyselektované hodnoty, granularita a aj zoradovanie.

Samotné triedenie prebieha v triede PrepareData v metóde SortData(). Doleuvedený pseudokód ukazuje časť tejto triedy, ktorá sa stará o triedenie podľa X-ovej osi.

**ak** je X-ová os nastavená ako časová

Skonvertuj položky X-ovej osi v pomocnej dátovej štruktúre na DateTime.  
Pomocou delegáta utried pomocnú dátovú štruktúru, podľa položiek X-ovej osi.

**inak**

Pomocou delegáta utried pomocnú dátovú štruktúru, podľa položiek X-ovej osi.

**koniec podmienky**

#### 4.3.8. Funkcia ZOOM a Full Screen

Obr.43 ukazuje metódu FullScreen, ktorá sa stará o prepnutie Silverlight aplikácie do režimu zobrazenia na celú obrazovku. Následne sa pri udalosti FullScreenChanged volá metóda ChangeScale, ktorá sa postará o to, aby systém vyplnil celú plochu obrazovky, presnejšie najväčšie možné miesto.

Nastáva však problém, ak je na stránke implementovaných viacero týchto systémov, ktoré používajú rovnaký XAP súbor. Pri prepnutí do Full Screen módu vyplní celú obrazovku iba prvý systém. Ostatné, z dôvodu, na ktorý sme zatiaľ nepišli, majú problém pri zistení veľkosti obrazovky. Riešením je, v prípade implementácie viacerých systémov na jednej stránke, použiť viacero XAP súborov.

```
private void FullScreen(object sender, RoutedEventArgs e)
{
    Application.Current.Host.Content.IsFullScreen =
        !Application.Current.Host.Content.IsFullScreen;
}
```

Obr.56: Metóda FullScreen

Na implementáciu funkcie ZOOM je použitý zaujímavejší postup. Využíva sa v ňom vlastnosť RenderTransform.

Táto vlastnosť umožňuje použiť objekty Transform na skosenie, rotáciu, posun elementu, alebo približovanie. Nižšie uvedený zoznam popisuje rôzne Transform objekty, ktoré môžu byť použité s vlastnosťou RenderTransform.

- RotateTransform: Rotuje objekty špecifikovaním uhla v stupňoch.
- SkewTransform: Skosí objekty o zadanú dĺžku v ose X alebo Y.
- ScaleTransform: Zväčší alebo zmenší objekty horizontálne, alebo vertikálne použitím mierky.
- TranslateTransform: Posunie objekty horizontálne alebo vertikálne o zadanú hodnotu.

Ešte existuje špeciálny typ transformácii, TransformGroup, pomocou ktorej je možné na jeden objekt použiť viacero transformácii v danom poradí. Napríklad sa môže nechať nejaký objekt rotovať a zmenšiť.

```
private void ZoomIn(object sender, RoutedEventArgs e)
{
    scale.ScaleX = scale.ScaleX * 1.05;
    scale.ScaleY = scale.ScaleY * 1.05;
}

<Grid x:Name="LayoutRoot">
    <Grid.RenderTransform>
        <ScaleTransform x:Name="scale" ScaleX="1" ScaleY="1" />
    </Grid.RenderTransform>
</Grid RowDefinitions>
```

Obr.57: Implementácia ZOOM

Obr.57 ukazuje implementáciu funkcionality ZOOM, za použitia Transform objektu ScaleTransform. Hlavný kontajner, v ktorom sú umiestnené všetky grafy obsahuje objekt ScaleTransform, ktorý má pri inicializácii nastavené hodnoty ScaleX aj ScaleY na 1. Pomocou metód ZoomIn a ZoomOut sa tieto premenné zväčšujú zmenšujú a tým vytvárajú efekt ZOOM-u.

#### 4.3.9. Save a Load

K ukladaniu a načítaniu grafov sa využíva IsolatedStorage. Jedná sa perzistentné dátové úložisko, ktoré je naviazané na počítač a aplikáciu. Dá sa k nemu pristupovať ako k virtuálnemu súborovému systému a ukladať doňho súbory, alebo ho využiť k ukladaniu dát systémom kľúč - hodnota.

Do IsolatedStorage sa dá uložiť približne 10 MB dát [9], pričom je ale možnosť túto hodnotu navýšiť. V takom prípade je potrebné povolenie od používateľa. Pravým kliknutím myši v Silverlight aplikácií

a následným vybratím príkazu Configuration sa dá v záložke Application Storage informovať o aplikáciách, ktoré IsolatedStorage využívajú.

System je postavený tak, aby na zobrazenie grafov stránkach stačil jeden .XAP súbor, ktorý vie obsluhovať viacero XML súborov. Teda viacero stránok používa odkaz na ten istý .XAP súbor, ale má rôzne XML dátové súbory. System musí pri ukladaní a nahrávaní brať do úvahy tento fakt a musí vedieť, ktorý súbor má nahráť, keďže ich môže byť viacero. System to rieši špeciálnym systémom pomenovávania súborov. Názov súboru, ktorý si system ukladá, je vlastne celá cesta k súboru, ktorý zobrazuje a je teda unikátny.

System využíva IsolatedStorage na uchovávanie XML súboru s dátami pre graf. Obr.45 ukazuje časť metódy, ktorá sa stará o Save.

```
using (IsolatedStorageFile store =
IsolatedStorageFile.GetUserStoreForApplication())
{
    IsolatedStorageFileStream stream =
        store.OpenFile(xmlName, FileMode.Create, FileAccess.Write);
    ThisDoc.Save(stream);
    stream.Dispose();
    MessageBox.Show("Grafy úspešne uložené");
    // datum ulozenia suboru
    IsolatedStorageFileStream streamDateTime =
        store.CreateFile(txtDatetimeName);
    StreamWriter writer = new StreamWriter(streamDateTime);
    writer.Write(DateTime.Now);
    writer.Close();
}
```

Obr.58: Implementácia Save

#### 4.3.10. Scale Break

O správne vykreslenie ScaleBreaku a pomocné výpočty sa stará hneď niekoľko metód v triede GridDraw.

Metóda *CountSegDownAndSegUp()* je najdôležitejšia a stará sa o samotné výpočty a naplnenie pomocných premenných. Vypočítava sa počet vodorovných segmentov, ktoré sa môžu následne vyhodit' a ďalšie pomocné premenné, ktoré sa sú potrebné pre správne vykreslenie ScaleBreaku. Nižšie je uvedený pseudokód metódy *CountSegDownAndSegUp()*.

pre každú hodnotu j Y-ovej osi každého stĺpca

pozri sa ako je j ďaleko od Y-ovej súradnice potenciálneho zobrazenia Scale Breaku a rozhodni, či sa má zmeniť hodnota v premennej max, alebo min

```
max <- hodnota veľkosti stĺpca najbližšie nad potenciálnou čiarou  
ScaleBreaku
```

```
min <- hodnota veľkosti stĺpca najbližšie pod potenciálnou čiarou  
ScaleBreaku
```

koniec cyklu

```
segDown <- počet segmentov pod čiarou ScaleBreak, ktoré môžu byť vyhodené
```

```
segDownAll <- počet všetkých segmentov pod čiarou
```

### 4.3.11. Implementácia do stránok

Samotná implementácia do stránok je naozaj veľmi jednoduchá a dala by sa zhrnúť do troch základných krokov.

1. Stiahnutie najnovšieho .XAP súboru tohto systému [10].
2. Vytvorenie XML súboru s nastaveniami a dátami pre graf(y).
3. Vložiť na stránky kód s odkazom (v absolútnom tvare) na .XAP a XML súbor.

Obr.59 ukazuje základný kód, ktorý treba vložiť do vhodne veľkého kontajnera, v ktorom sa má zobrazovať. Napríklad v DIV elemente.

```
<object data="data:application/x-silverlight-2,"  
type="application/x-silverlight-2" width="1050" height="600">  
<param name="source" value="GraphicChart.xap"/>  
<param name="minRuntimeVersion" value="2.0.31005.0" />  
<param name="autoUpgrade" value="true" />  
<param name="initParams"  
value="xmlSource=http://chart.4webdesign.sk/data.xml" />  
  
<a href="http://go.microsoft.com/fwlink/?LinkID=124807"  
style="text-decoration: none;">  
  
</a>  
</object>
```

Obr.59: Object implementujúci tento systém

Na implementáciu je použitý HTML element *object*. Ako vidieť, samotná implementácia do stránok je veľmi jednoduchá a kód netreba ani veľmi popisovať. Width a height sú vyžadované hlavne kvôli správne mu pracovaniu Full Screen módu. Taktiež treba vhodne nastaviť tieto hodnoty, aby sa do

tohto objektu grafy vošli. Štvrtý riadok je odkaz na .XAP súbor a v šiestom riadku sa nachádza odkaz na dátový XML súbor.

Pre správne fungovanie systému je potrebné dodržiavať nasledujúci postup.

1. v XML súbore musia byť korektne vypísané všetky tagy. Ak sa graf na stránke nezobrazuje, s veľkou pravdepodobnosťou nebol niektorý tag správne vypísaný, prípadne sa na niektorý zabudlo.
2. Pri vypisovaní tagov width a height v XML súbore, treba brať ohľad na to, že kvôli popiskom, tlačidlám a ľavému menu bude samotný graf menší. Treba rátať s rezervou približne 110px na šírke aj výške.  
Preto pri zadaní width=500 a height=300 bude reálna veľkosť grafu asi 390px x 190px
3. Treba voliť vhodnú dĺžku textov, ktoré sa budú zobrazovať ako popisky. Môže totiž nastať situácia, kedy sa nevojdú do grafov, presnejšie časť z nich nebude vidieť, keďže priestor na zobrazenie grafov je obmedzený. Preto treba voliť čo najkratšie a najvýstižnejšie názvy, vid'. ukážky [10].
4. Treba správne zadať cestu k XML súboru, prípadne ZIP súboru v absolútnom tvare. Taktiež treba zadať správnu cestu ku XAP súboru.
5. V prípade použitia viacerých týchto systémov na jednej stránke je kvôli zachovaniu 100% funkčnosti Full Screen módu potrebné, aby mal každý systém svoj vlastný XAP súbor. Viac v kapitole 4.3.8.

## 5. Záver

V diplomovej práci sa podarilo splniť ciele, ktoré boli na začiatku vytýčené. Urobil som prehľad dostupných riešení a technológií, pričom poznatky, ktoré som nadobudol, som následne preniesol do implementácie vlastného systému. Navrhol som interaktívny systém umožňujúci vykresľovať viacero grafov súčasne, pričom je veľmi jednoducho zakomponovateľný do stránok. Taktiež som vytvoril web so systémom, ukázkami, ako aj dokumentáciou na stiahnutie [10].

Systém by som chcel po obhajobe tejto diplomovej práce rozšíriť o ďalšiu funkcionality. Napríklad vykresľovanie iných typov grafov, prerobenie SAVE a LOAD, aby sa dalo ukladať viacero grafov, vytvorenie šablónovacieho systému, ktorý by ľahko umožnil meniť výzor grafov,... Systém by som rád ďalej vylepšoval s ohľadom na lepšiu a intuitívnejšiu užívateľskú prívetivosť a interakciu s užívateľom v duchu Web 2.0 aplikácií.

Skúsenosti, ktoré som nadobudol počas vytvárania tejto práce, považujem za neoceniteľné do budúceho profesionálneho života.



## 6. Zoznam použitej literatúry

- [1] Jeff Scanlon: Accelerated Silverlight 2, 2008 Apress
- [2] Matthew MacDonald: Pro Silverlight 2 in C# 2008, 2008 Apress
- [3] Jit Ghosh, Rob Cameron: Silverlight 2 Recipes, 2009 Apress
- [4] Matthew MacDonald: Silverlight 2 Visual Essentials, 2008 Apress
- [5] Christian Wenz: Essential Silverlight 2, 2008 O'Reilly
- [6] Chad Campbell, John Stockton: Silverlight 2 in Action, 2008 Manning Publications
- [7] Christian Nagel, Bill Evjen, Jay Glynn, Morgan Skinner, Karli Watson, Allen Jones: C# 2005 Programujeme profesionálne, 2006 Computer Press
- [8] Steve Krug: Web design Nenúťte užívateľa premýšľať, 2003 Computer Press
- [9] Ľuboslav Lacko: Silverlight 2, 2008 Microsoft
- [10] <http://www.4webdesign.sk/chart.html>
- [11] MSDN: Microsoft Developer Network  
<http://msdn.microsoft.com/sk-sk/default.aspx>
- [12] Charts And Graphs: Modern Solutions, 2007 Smashing Magazine  
<http://www.smashingmagazine.com/2007/10/18/charts-and-graphs-modern-solutions/>
- [13] Ján Hanák: Objektovo orientované programovanie v jazyku C# 3.0, 2008 Microsoft
- [14] Visifire dokumentácia  
[http://www.visifire.com/visifire\\_charts\\_documentation.php](http://www.visifire.com/visifire_charts_documentation.php)
- [15] Fabio Claudio Ferracchiati: LINQ for Visual C# 2008, 2008 Apress
- [16] Scott Klein: Professional LINQ, 2008 Wiley Publishing