

Technologie WPF a Silverlight

WPF and Silverlight technologies

Bc. Tomáš Petura



*** nascannované zadání str. 1 ***

*** nescannované zadání str. 2 ***

ABSTRAKT

Obsahem této diplomové práce je rešerše literárních zdrojů na téma WPF a Silverlight, rozbor o výhodách těchto technologií, jejich využití v konkrétních aplikacích a studijní opora pro studenty zájímající se o tyto technologie spolu s testovými otázkami, pro ověření jejich znalostí.

Klíčová slova: WPF, Silverlight, Visual Studio, Expression Blend, Microsoft, XAML, C#, VB

ABSTRACT

The content of these thesis is literature search on WPF and Silverlight. Analysis of the benefits of these technologies, their use in the specific applications and educational support for students who are interested in these technologies, with test questions to verify their knowledge.

Keywords: WPF, Silverlight, Visual Studio, Expression Blend, Microsoft, XAML, C#, VB

Děkuji svému vedoucímu Ing. Bc. Pavlovi Vařachovi, Ph.D., za odbornou pomoc a čas, který mi věnoval při realizaci této diplomové práce.

Prohlašuji, že

- beru na vědomí, že odevzdáním diplomové/bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že diplomová/bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk diplomové/bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji diplomovou/bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – diplomovou/bakalářskou práci nebo poskytnout licenci k jejímu využití jen s předchozím písemným souhlasem Univerzity Tomáše Bati ve Zlíně, která je oprávněna v takovém případě ode mne požadovat přiměřený příspěvek na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše);
- beru na vědomí, že pokud bylo k vypracování diplomové/bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky diplomové/bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem diplomové/bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

- že jsem na diplomové práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze diplomové práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně

.....
podpis diplomanta

OBSAH

ÚVOD.....	10
I TEORETICKÁ ČÁST	11
1 WINDOWS PRESENTATION FOUNDATION	12
1.1 XAML	12
1.2 VEKTOROVÁ GRAFIKA	12
1.3 VISUAL TREE A LOGICAL TREE	13
1.3.1 Logical Tree	14
1.3.2 Visual Tree	15
2 SILVERLIGHT	16
2.1 DESKTOPOVÉ VS. WEBOVÉ APLIKACE	16
2.2 HISTORIE SILVERLIGHTU	17
2.2.1 Silverlight 2	18
2.2.2 Silverlight 3	19
2.2.3 Silverlight 4	19
2.2.4 Silverlight 5	19
3 SILVERLIGHT VS. WPF	21
4 REŠERŠE.....	22
4.1 BEGINNING SILVERLIGHT 4 IN C#	22
4.2 INTRODUCING SILVERLIGHT 4	23
4.3 PRO SILVERLIGHT 5 IN C#	24
4.4 PRO SILVERLIGHT 5 IN VB	25
4.5 PRO WPF IN C# 2010: WINDOWS PRESENTATION FOUNDATION IN .NET 4.....	26
4.6 PROGRAMING WPF	27
4.7 PROFESSIONAL WPF PROGRAMMING: .NET DEVELOPMENT WITH THE WINDOWS PRESENTATION FOUNDATION	28
4.8 PRO WPF IN VB 2010: WINDOWS PRESENTATION FOUNDATION IN .NET 4.....	29
4.9 FOUNDATION EXPRESSION BLEND 4 WITH SILVERLIGHT	30
4.10 PRO EXPRESSION BLEND 4	31
4.11 MICROSOFT DEVELOPER NETWORK SUBSCRIPTION LIBRARY (MSDN LIBRARY)	31
4.12 STACK OVERFLOW	32
4.13 THE CODE PROJECT.....	33
5 SILVERLIGHT V PRAXI	34

5.1	SILVERLIGHT NA OLYMPIJSKÝCH HRÁCH	34
5.2	TESCO SW A.S. - FAMA+	36
5.3	BING MAPS.....	39
6	WPF APLIKACE	41
6.1	VISUAL STUDIO 2010	41
6.2	FAMILY.SHOW.....	41
II	PRAKTICKÁ ČÁST	42
7	SILVERLIGHT VS FLASH VS JAVA.....	43
7.1	JAVAFX	43
7.2	ADOBE FLASH	43
7.3	VYUŽÍVÁNÍ TECHNOLOGIÍ PRO RIA APLIKACE.....	44
7.4	HTML 5.....	46
8	PROČ WPF	47
8.1	QT 47	
8.2	ROZŠÍŘOVÁNÍ WPF	47
8.3	WPF VS WINDOWS FORMS.....	48
9	TESTOVACÍ OTÁZKY	50
9.1	DESERIALIZACE.....	50
9.2	TESTOVACÍ APLIKACE.....	51
9.3	UKÁZKA TESTOVÝCH OTÁZEK	51
10	STUDIJNÍ OPORA.....	53
10.1	HELLO WORD.....	53
10.2	STUDIJNÍ OPORA	56
10.2.1	Vytvoření třídy Osoba	56
10.2.2	Tvorba grafického rozhraní	58
10.2.3	Codebehind MainWindow	59
10.2.4	Binding.....	60
10.2.5	Serializace a Deserializace	61
10.2.5.1	Serializace	62
10.2.5.2	Deserializace	63
10.2.6	Binding 2.....	64
10.2.7	Stylování.....	65
	ZÁVĚR	68
	ZÁVĚR V ANGLIČTINĚ.....	69
	SEZNAM POUŽITÉ LITERATURY.....	70
	SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK	73
	SEZNAM OBRÁZKŮ	74

SEZNAM TABULEK.....	76
----------------------------	-----------

ÚVOD

Cílem diplomové práce bylo vytvořit rešerši literárních a elektronických zdrojů na téma WPF a Silverlight tak, aby pokryla rozličné zdroje pro studium obou těchto technologií a usnadnila orientaci v širokém množství dostupných knih věnující se této problematice.

Dalším bodem bylo analyzovat možnosti WPF a Silverlightu v komerční sféře, jejich použití a rozšíření. Také bylo úkolem uvést konkrétní využití těchto technologií.

Posledním úkolem bylo vytvořit studijní oporu pro studenty předmětu Technologie .NET 4, která jim přiblíží možnosti této technologie, spolu s testovacími otázkami, které ověří jejich znalosti o technologiích WPF a Silverlight.

V teoretické části jsou nejprve uvedeny základní informace o technologii WPF a technologii Silverlight, spolu s uvedením rozdílů mezi těmito dvěma technologiemi. Následuje rešerše dostupné literatury a to, jak elektronických tak i klasických knih. Jsou zde popsány zdroje popisující obě technologie s využitím VB tak i C#. V poslední dvou kapitolách teoretické části jsou zmíněny aplikace vytvořené pomocí těchto dvou technologií.

Praktická část diplomové práce začíná srovnáním technologií WPF a Silverlight s jejich konkurenty. V případě Silverlightu se jedná o Adobe Flash a JavaFX, u WPF je to Windows Forms a QT. Následuje popis řešení otázek pro ověření znalostí studentů a v poslední části diplomové práce je studijní opora pro studenty, která poskytuje čtenářům základní znalosti o těchto technologiích na konkrétních příkladech.

I. TEORETICKÁ ČÁST

1 WINDOWS PRESENTATION FOUNDATION

Když byl poprvé představen .NET Framework, byl jeho součástí i Windows Form, což je plně vybavený toolkit, který obsahuje všechny potřebné prvky pro navržení standardního vzhledu aplikace. Obsahuje základní prvky jako jsou tlačítka, textová pole, checkbox atd. Výsledkem ovšem je nemožnost jednoduché úpravy vzhledu. Pokud byste chtěli vytvořit například zářící nebo jiný efekt okolo tlačítka, museli byste vytvořit vlastní custom control a vytvořit vzhled tlačítka pro každý stav, ve kterém se může nacházet (reakce na najetí myši, stisknutí tlačítka myši atd.). [3]

Windows Presentation Foundation (WPF) změnil tento zaběhnutý model, ačkoliv obsahuje standardní ovládací prvky. Každý tento prvek obsahuje vlastní text, border a background, o jejichž vykreslení se stará samotná komponenta. Výsledkem je, že WPF je mnohem mocnější nástroj pro úpravu vzhledu než jeho předchůdce Windows Form. Používáním těchto vlastností můžete stylovat všechny ovládací prvky, bez toho aniž byste museli psát jakýkoliv kód, což umožňuje tvorbu graficky unikátních aplikací. Vzhled komponent a jejich grafická reprezentace v jednotlivých stavech je uložena ve formátu XAML. [3]

1.1 XAML

Dle definice je XAML značkovací jazyk pro specifikaci a nastavení vlastností tříd, založený na XML. Jinými slovy: XAML je jazyk používaný ve WPF, Silverlightu nebo kdekoli jinde, kde si potřebujete sami deklarovat třídy. S jeho pomocí můžete deklarovat proměnné a nastavovat vlastnosti pro všechny třídy ve vaší aplikaci. XAML parser automaticky vytvoří objekt při spuštění aplikace podle definice v XAML. [21]

XAML je hlavně používán pro návrh uživatelského rozhraní a tvorbu objektů. Prostřednictvím XAML ovšem nemůžete definovat chování aplikace. Z toho plyne, že i když má XAML široké možnosti, není to ve své podstatě programovací jazyk, ale používá se hlavně pro návrh uživatelského rozhraní aplikace. Proto XAML používá jiné programovací jazyky jako je C# nebo VB pro tvorbu výkonného kódu. [21]

1.2 Vektorová grafika

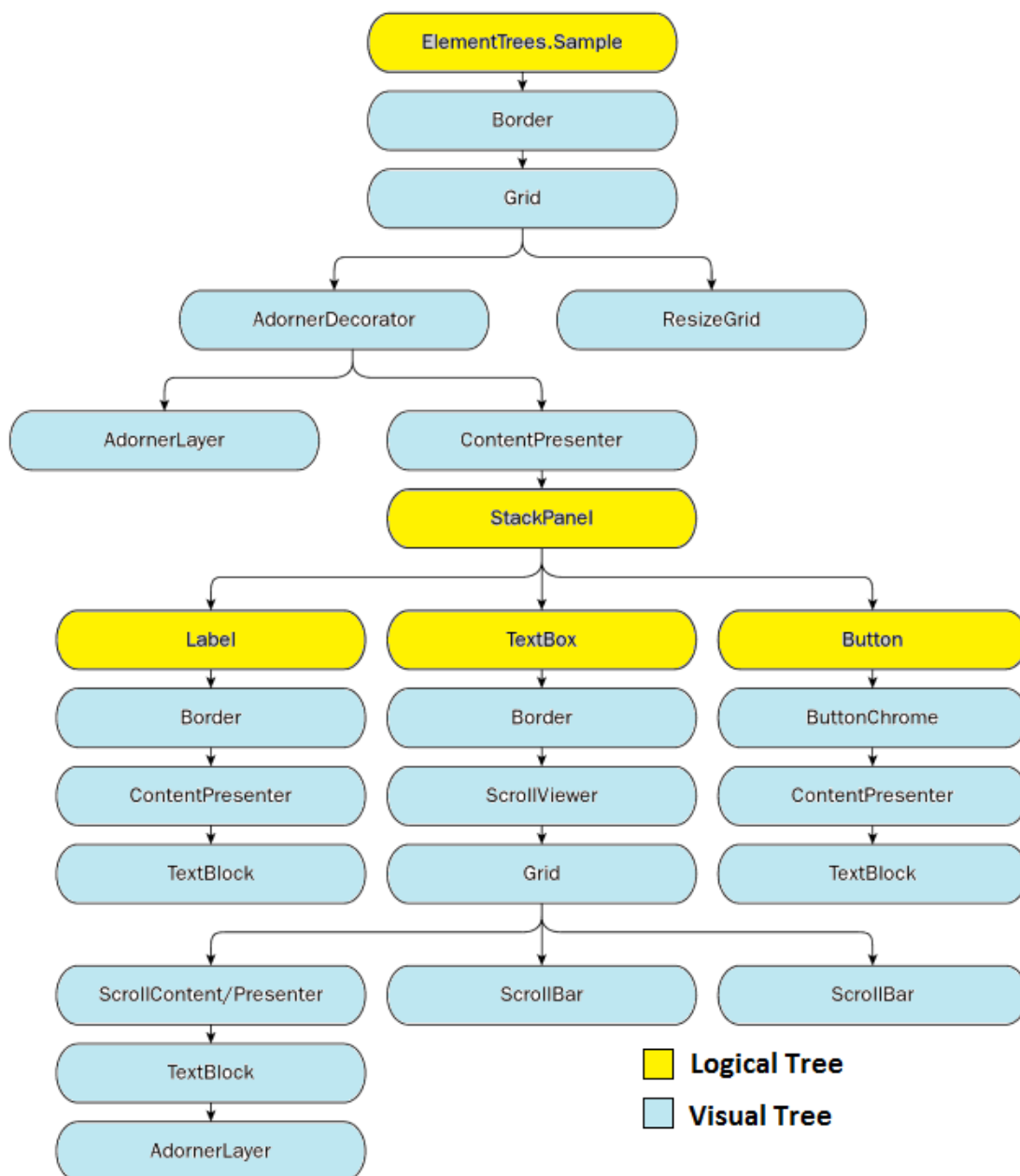
Aby bylo ve WPF využito nových, výkonných grafických karet, engine využitý v této technologii podporuje vektorovou grafiku. To umožňuje například zoomování, aniž by se

zhoršila kvalita obrazu, což nebylo dříve z důvodu využívání rastrové grafiky možné. Jako prostředek k renderování vektorové grafiky je využíván Direct3D. [5]

1.3 Visual Tree a Logical Tree

Pro pochopení, jak jsou prvky ve WPF navzájem propojeny je potřeba vysvětlit dva pojmy: Visual Tree a Logical Tree.

Prvky grafického rozhraní ve WPF jsou hierarchicky skládány. Tento vztah je označován jako Logical Tree. Template jednoho logického prvku obsahuje několik vizuálních prvků. Tento strom se potom nazývá Visual Tree. WPF rozlišuje mezi těmito dvěma stromy, protože v některých případech může využívat pouze logických prvků a v jiných jsou potřeba oba typy prvků. [8]



Obr. 1. Visual Tree

Pro každý prvek logického stromu mohou být vytvořeny další, dodatečné prvky, sloužící ke grafické reprezentaci prvku. Kombinací logických a vizuálních prvků vzniká Visual Tree. [5]

1.3.1 Logical Tree

Popisuje vztahy mezi prvky uživatelského rozhraní.

Logical Tree je zodpovědný za:

- Dědění Dependency Property

- Vyhodnocování Dynamic Resources
- Binding
- Posílání Routed Events [8]

1.3.2 Visual Tree

Obsahuje logické a vizuální prvky všech šablon (template).

Visual Tree se stará o:

- Vykreslování vizuálních prvků
- Nastavování průhlednosti prvku
- Provedení transformace (např. otočení, zoom)
- Nastavení property IsEnable
- Hit - Testing [8]

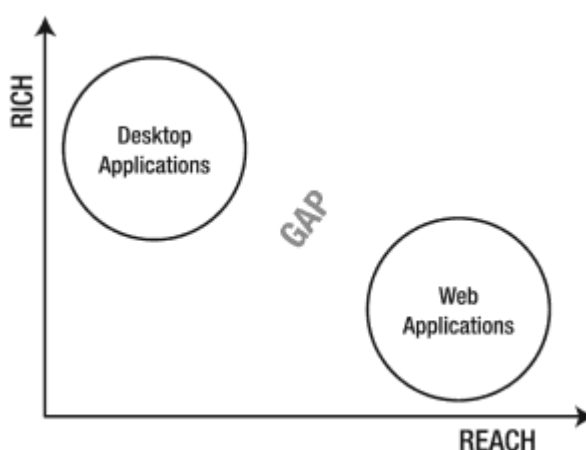
2 SILVERLIGHT

Microsoft Silverlight je technologie založená na Microsoft .NET Frameworku, určená pro vývoj next-generation aplikací, aplikací s bohatým multimediálním obsahem a rich internet applications (RIAs), které jsou nezávislé na platformě, zařízení a prohlížeči. Multiplatformní aplikace postavené na Silverlightu můžou zajistit maximální spokojenost zákazníků a pomáhá společnosti maximalizovat vloženou investici. [2]

2.1 Desktopové vs. webové aplikace

Desktopové aplikace nabízejí bohaté možnosti grafické kustomizace, ale mají horší dostupnost než webové aplikace. Takové aplikace musejí být naprogramovány pro všechny cílové platformy zvlášť a každý stroj, na kterém je tato aplikace nainstalována, musí být udržován. [1]

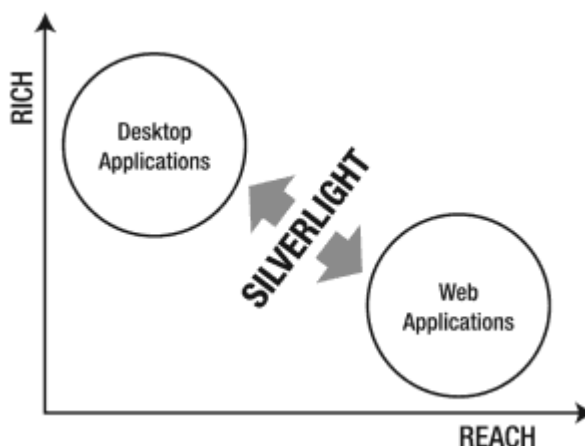
Na druhé straně jsou webové aplikace, které je možné spouštět na různých platformách. Vývojáři zaměřeni na technologii .NET Framework, byli dříve nuceni využívat ASP.NET a poskytovat aplikace prostřednictvím internetových služeb. Znamenalo to přesun většiny logiky a kódu na server ve prospěch snížení nároků aplikací na výkon počítače. Cenou ovšem bylo chudé uživatelské rozhraní. Tyto aplikace měly vynikající dostupnost, ale špatné uživatelské rozhraní. [1]



Obr. 2. Desktopové vs Webové aplikace

K zaplnění této mezery bylo potřeba vyvinout nový přístup k vývoji aplikací, označovaný jako Rich Internet Applications (RIA). RIA je definován jako webová aplikace, které má

vlastnosti a funkcionalitu obvyklou u tradičních desktopových aplikací. Existuje několik RIA technologií, jednou z nich je právě Microsoft Silverlight. [1]



Obr. 3. Silverlight aplikace

2.2 Historie Silverlightu

Informací o Silverlightu před jeho prvním oficiálním představením na konferenci MIX07 v březnu 2007 bylo velmi málo. V té době se o něm mluvilo pouze jako o WPF/E (Windows Presentation Foundation / Everywhere). I když informací, které unikly na veřejnost nebylo mnoho, jedno bylo jasné: jedná se o WPF hostované v prohlížeči. [2]

V době oficiálního představení Silverlight 1 ještě nepodporoval CLR a jeho součástí nebylo ani mnoho dalších vlastností, které přišli až s verzí Silverlight 2. Co ovšem obsahoval, byla základní podmnožina funkcí XAML a nastínění budoucích možností technologie Silverlight. Nejzřetelnější možností Silverlightu 1 byla možnost napsání aplikace čistě v XAML nebo kombinací XAMLu s JavaScriptem a pomocí Document Object Model (DOM) upravovat uživatelské rozhraní. Co se nezměnilo od první verze Silverlightu, je nutnost mít nainstalován plug-in na straně klienta. [2]

Hlavními rysy Silverlightu 1 byly:

- Základ architektury: Tím jsou DependencyObject-y a UIElement-y, které tvoří základní třídy uživatelského rozhraní.
- Basic layout: Tím byl Canvas, umožňoval umisťovat prvky pouze pomocí absolutní hodnoty.

- Základní ovládací prvky: TextBlock a další prvky umožňovali zobrazování textu. Pro vkládání vstupů uživatelem zde nebyly žádné komponenty. Tato možnost přibyla v beta verzi Silverlightu 2.
- 2D grafika: Třídy založené na geometrických objektech, umožňující vykreslování 2D tvarů.
- Média: První aplikace postavené na Silverlightu ukazovali možnosti videa a obrázků. Součástí byla podpora jednoduchého stažení mediálních prvků tak, aby byla škála využití této technologie co nejširší. Silverlight Media Player podporoval WMA, WMV a MP3 formáty.
- Animation: Třída Storyboard z WPF byla součástí XAMLu již od první verze Silverlightu, umožňovala animace prvků uživatelského rozhraní nejrozličnějšími způsoby.
- Brushes a transformace: Image brush, video brush a color brush (což je nastavování barev a sklonů) byli součástí Silverlightu od první verze.

Dvě nejdůležitější vlastnosti Silverlightu přišli až v pozdějších verzích a tím byla široká škála ovládacích prvků a integrace code-behind, což je funkční vrstva za XAML definicí GUI a doplňuje jej o funkce jazyka C# nebo Visual Basic. [2]

2.2.1 Silverlight 2

Těsně před konferencí MIX08 v březnu 2008, Microsoft odhalil verzi Silverlightu 1.1. Ta byla nakonec označena jako verze 2, protože obsahovala velké množství změn.

Zásadní změny v Silverlightu 2:

- Silverlight 2 byl založen na .NET Frameworku 3.5
- Toolkit pro Silverlight měl více než sto komponent (Grid, Canvas, ListBox, DataGrid, TextBox, ScrollViewer a další)
- Podpora CLR
- ADO.NET, LINQ, LINQ to XML a X.Linq
- Podpora grafických prvků a animací: 2D vektorová grafika, zoomování
- Podpora asynchronní komunikace a vedlejších vláken [2]

2.2.2 Silverlight 3

Microsoft pokračoval v tradici uvádění nových verzí a novou verzí vydal opět v průběhu konference MIX09. V které se Microsoft zaměřil především na grafické možnosti Silverlightu.

Novinky v Silverlightu 3 oproti předešlé verzi byly:

- Podpora 3D grafiky
- Podpora témat (Theme) - umožňuje změnu vzhledu aplikace "runtime"
- Podpora nového formátu H.264 / Advanced Audio Coding / MP4
- Nový Layout - DockPanel, WrapPanel a TabPanel
- Nová formulářová okna - PasswordBox, AutoCompleteBox, SaveFileDialog a Save-As dialog [2]

2.2.3 Silverlight 4

Verze Silverlight 4 byla zveřejněna v dubnu 2010 na konferenci DevConnection. Microsoft si uvědomil, že pro Line of Business (LoB) aplikace je důležitá podpora tisku, integrace dokumentů, reportingu, podpora práce s lokálními zařízeními a možnost práce v offline režimu.

Novinky ve verzi Silverlight 4:

- Podpora tisku
- Kontextové menu pro pravé tlačítko myši, jako u desktopových aplikací
- Komponenta pro formátování textu RichTextBox
- Možnost spustit Silverlight aplikaci v režimu out-of-browser. Což umožnilo vystupovat Silverlight aplikacím jako důvěrné
- Přístup aplikace do "Mých" souborů (Dokumenty, Videá, Obrázky a Hudba) [2]

2.2.4 Silverlight 5

Ve verzi Silverlight 5 se Microsoft zaměřil hlavně na využití výkonu. Hlavními změnami jsou:

- Využití výkonu: rychlejší spouštění Silverlight 5, podporuje 64-bitové prohlížeče a byla zlepšena čitelnost malých písmen
- Vektorový tisk: Byly rozšířeny možnosti tisku o vektorový tisk. Výsledkem je rychlejší tisk s menším využitím paměti
- Hardwarová akcelerace 3D grafiky
- Trick play: Nová funkcionalita, která umožňuje zpomalené / zrychlené přehrávání videa bez změny zvukové stopy. Například přehrávání záznamu z přednášek
- Dvojklik a trojklik myši: Sice se jedná o malou změnu, ale řeší řadu nepříjemností
- XAML debugging: Umožňuje umístit breakpoint na data binding
- Child window: Aplikace běžící v režimu out-of-browser umožňují otevření dalšího okna jako klasické Windows aplikace
- Úplný přístup k souborům: Aplikace, které jsou spuštěné jako důvěryhodné mají nyní přístup ke všem souborům, ke kterým má přístup uživatel
- Důvěryhodná aplikace spuštěná v prohlížeči: Jedná se o novou skupinu aplikací kombinující výhody důvěryhodných aplikací a klasických Silverlight aplikací spuštěných v prohlížeči. Toto nastavení je možné pouze v uzavřeném prostředí (například v intranetu společnosti), kde jsou nainstalované potřebné certifikáty [7]

3 SILVERLIGHT VS. WPF

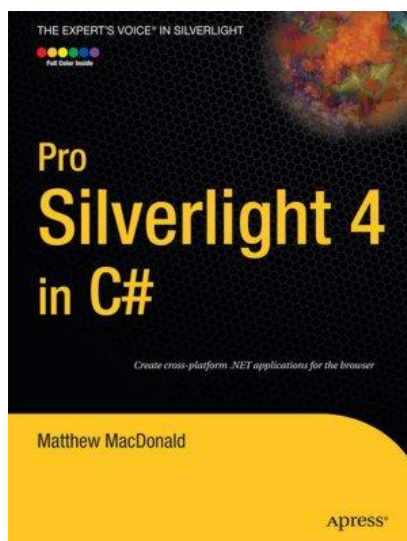
Silverlight byl před oficiálním vydáním znám pouze pod označením Windows Presentation Foundation Everywhere (WPF/E), je to multiplatformní verze WPF. V podstatě se jedná o podmnožinu WPF, která může být spuštěna jak na Mac-u, tak i na PC. Oproti tomu WPF je spustitelné pouze na Windows 7, Windows Vista nebo Windows XP s nainstalovaným .NET 3.0 nebo vyšším (Windows XP je defaultně dodáván s .NET 2.0). Jedním z nejdůležitějších rozdílů je to, že Silverlight je přímo součástí HTML stránky a je spouštěn pomocí nainstalovaného plug-inu, zatímco WPF musí být vložen do iframe, aby ho bylo možné zobrazit uvnitř HTML stránky. Do verze Silverlight 5 byl jedním ze zásadních rozdílů také způsob vykreslování 3D objektů. Zatím co WPF skutečně vykreslovalo 3D objekty a využívalo k tomu grafickou kartu, Silverlight pouze předstíral, že se jedná o 3D objekt.

4 REŠERŠE

Jak k Silverlightu, tak i k WPF je k dispozici dostatečné množství literatury a to jak v tištěné formě tak i ve formě elektronické. K oběma technologiím vychází dva druhy knih rozdělných podle programovacího jazyka, který je použit k psaní funkčního kódu (code behind), kterým může být C# nebo Visual Basic. U Silverlightu dochází navíc k častému vydávání nových verzí a proto mnozí autoři aktualizují své knihy vždy pro nově vydanou verzi Silverlightu.

Všechny knihy, zmiňované v další části předpokládají, že má čtenář alespoň základy programování v jazyce C#, respektive Visual Basic, samy o sobě jsou věnovány samotné tvorbě aplikací prostřednictvím WPF a Silverlightu.

4.1 Beginning Silverlight 4 in C#



Obr. 4. Beginning Silverlight 4 in C#

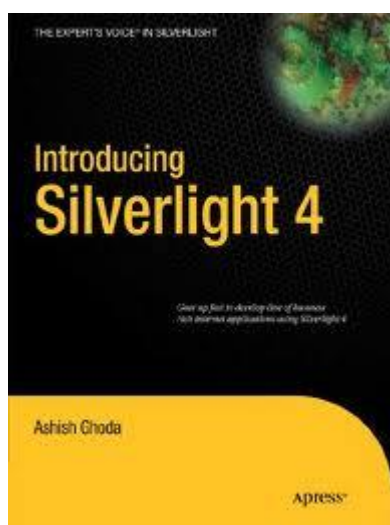
Robert Lair [1]. Na začátku knihy autor představí Silverlight, jeho vývoj od prvních verzí po současnost a popíše jeho výhody oproti klasickým desktopovým aplikacím. V jednotlivých kapitolách autor vysvětlí data binding a layout management, popíše ovládací prvky, které můžete použít při vytváření vlastních aplikací a jejich stylování. V jedné z posledních kapitol se zaměřuje na animace, jedna z hlavních výhod Silverlightu a odlišuje RIA od klasických internetových aplikací.

V knize je také věnována samostatná kapitola programu Expression Blendu. Jako základ a představení možností tohoto nástroje je kapitola svým obsahem dostatečná. Pokud byste se

ale chtěli dozvědět více o tom, jak vytvářet vlastní komponenty a hlavně jak upravovat jejich vzhled, je vhodnější literatura, která je zaměřená přímo na program Expression Blend.

V celé knize jsou uvedeny příklady a zpětná vazba na to, jak byla daná problematika řešena v předchozích verzích Silverlightu, což dává představu o vývoji této technologie.

4.2 Introducing Silverlight 4



Obr. 5. Introducing Silverlight

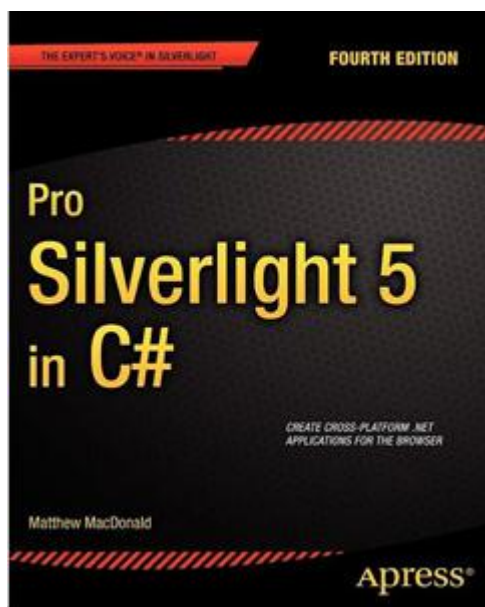
Ashish Ghoda [2]. Několik prvních kapitol knihy se nevěnuje tvorbě vlastní aplikace, ale je zaměřeno na samotný Silverlight, jako na technologii, na jeho historii a vývoj. Tyto kapitoly jsou zajímavé hlavně proto, že umožňují nahlédnout jak se Silverlight formoval do podoby v jaké ho známe dnes.

Velmi podrobně jsou v knize popsány jednotlivé komponenty grafického rozhraní. Jsou rozděleny do čtyř skupin (Layout Management, formulářové komponenty, komponenty pro práci s daty a funkční prvky). Každé komponentě je věnována samostatná kapitola, popisující její vlastnosti (property) a u většiny i ukázkou použití.

V dalších kapitolách se autor věnuje stylování, grafice nebo práci s vlákny. Za zmínku stojí část knihy věnovaná funkcionalitě Out-of-Browser, kde autor například vysvětluje, jak tuto aplikaci vytvořit, nainstalovat nebo aktualizovat verze.

Jako materiál pro základy programování v Silverlightu tato kniha vhodná není, jedná se spíše o knihu zaměřenou na technologii Silverlight. Například binding je zde popsán velmi krátce, stejně tak i Dependency Properties.

4.3 Pro Silverlight 5 in C#



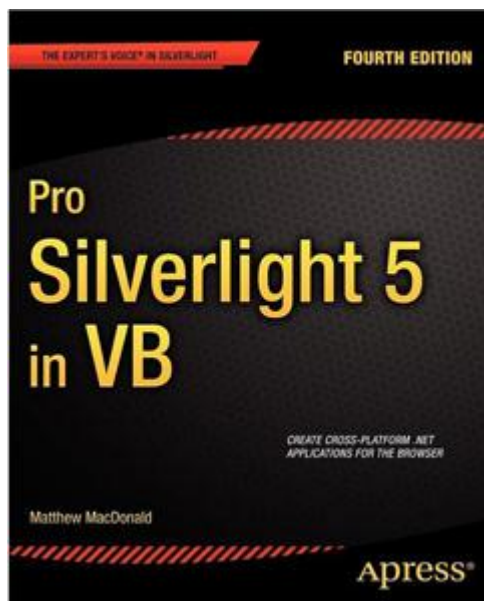
Obr. 6. Pro Silverlight 5 in C#

Matthew MacDonald [7]. Jedná se o jednu z prvních knih věnující se Silverlight 5. Hned v úvodu se autor pouští do srovnání této technologie s jeho konkurenty Flash, HTML5.

Pokud již máte k dispozici knihu o Silverlight 4, tak v této knize se nedozvíte příliš nových informací, protože většina věcí zůstala stejná jako v předchozí verzi. Pokud ovšem máte zájem o využití 3D efektů nebo vytváření Out-of-Browser aplikací, je zde několik užitečných informací. Verze Silverlight 5 umožňuje využít hardwarové akcelerace při vykreslování 3D objektů a také rozšiřuje možnosti vytvoření důvěryhodných aplikací.

Skladba ostatních kapitol je podobná jako u ostatních knih, autor se věnuje data bindingu, stylování a jednotlivým komponentám. Podrobněji se rozepisuje o animacích a práci se zvukem, videem, zoomování a přístupu k souborům.

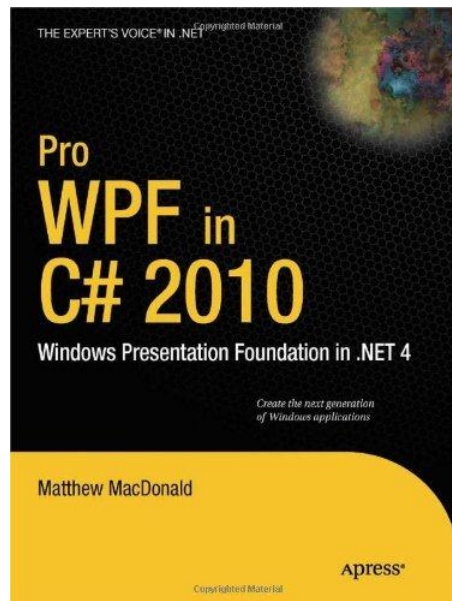
4.4 Pro Silverlight 5 in VB



Obr. 7. Pro Silverlight 5 in VB

Matthew MacDonald [19]. Kniha podrobně popisuje široké možnosti Silverlightu včetně jeho nové verze Silverlight 5. Na rozdíl od ostatních knih popisuje kombinaci XAML a VB. Jedná se o knihu, která se tematikou zabývá velmi podrobně a čtenáře provede možnostmi Silverlightu od úplného začátku. Postupně je čtenář seznámen s XAML, layouty, dependenci property, animacemi, multimediálními možnostmi Silverlightu (zvuk, video, zoom), styly nebo Out-of-Browser aplikacemi. Na rozdíl od knih věnující se Silverlight 4 je zde široce popsána možnost 3D vykreslování. Je to pravděpodobně z toho důvodu, že ve verzi Silverlight 5 se již jedná o skutečné 3D objekty a nejen o simulování 3D efektu na 2D objektech a také díky širšímu využití hardwarového výkonu při vykreslování grafiky.

4.5 Pro WPF in C# 2010: Windows Presentation Foundation in .NET 4



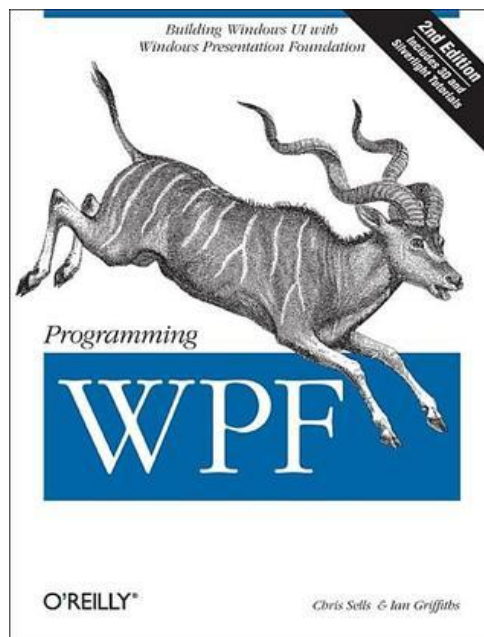
Obr. 8. Pro WPF in C# 2010:WPF in .NET 4

Matthew MacDonald [3]. V knize je vše popsáno velmi podrobně a srozumitelně, pro pochopení obsahu stačí znalost jazyka C#, předchozí zkušenosti s WPF nejsou potřeba. Autor se dopodrobna věnuje ve všech kapitolách vybranému tématu a kniha může přinést nové informace i pokročilejším programátorům.

Dobře je zpracována kapitola věnující se Dependency Properties. Možnostem vykreslování a animacím je zde také věnován dostatek prostoru.

Na konci každé kapitoly je sekce "The Last Word", ve které autor rekapituluje probranou kapitolu, upozorní na důležité pojmy a zopakuje důležité informace, což umožňuje čtenáři zkontrolovat, zda pochopil probranou látku a případně se vrátit k tomu co přehlédl.

4.6 Programing WPF

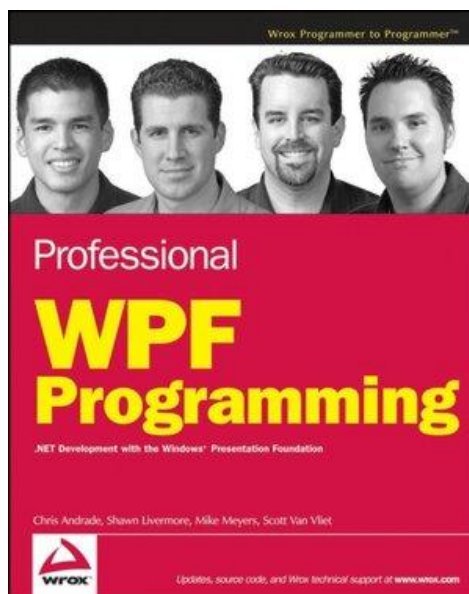


Obr. 9. Programing WPF

Chris Sells, Ian Griffiths [4]. I když se jedná o starší knihu, ve které je tématika WPF vysvětlována ve Visual Studio 2005 a .NET Framework 3.0, je vhodná jako úvod do WPF, ideálně kombinovaná s jinou literaturou, která bude zdrojem novinek v .NET Frameworku 4.0. Autoři na začátku upozorňují, že u čtenářů předpokládají znalost programování v C# a .NET a výhodou je znalost Windows Forms, XML a HTML.

Nevýhodou je, vzhledem ke staršímu datu vydání, absence jakékoliv zmínky o Microsoft Expression Blend. Kromě tohoto nedostatku je v knize popsáno vše potřebné a pokud by autoři vydali novou edici pro Visual Studio 2010 a .NET Framework 4.0, jednalo by se o jednu z nej kvalitnějších knih o WPF.

4.7 Professional WPF Programming: .NET Development with the Windows Presentation Foundation

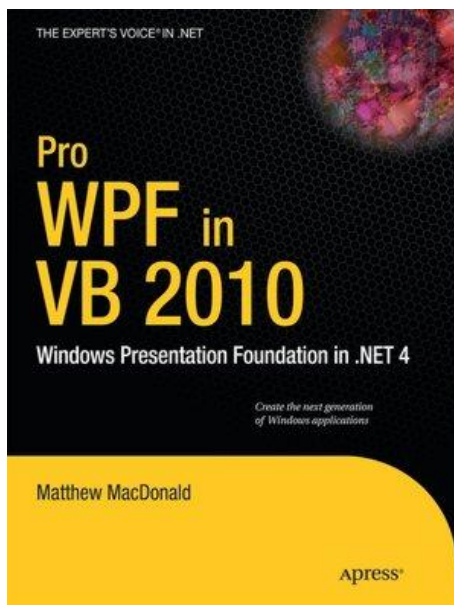


Obr. 10. Professional WPF Programmin

Chris Andrade, Shaw Livermore, Mike Meyers, Scott Van Vliet [5]. Jednou z hlavních výhod a odlišností od ostatních knih je velikost prostoru, který je věnován možnostem stylování komponent s využitím Microsoft Expression Blend, jenž bývá v ostatní literatuře jen zmíněn. Zde jsou mu věnovány dvě kapitoly.

Dalším tématem, na které v ostatních knihách nenarazíte nebo mu nebyl věnován dostatek prostoru je propojení WPF a Win32 aplikací a to jak použití WPF ve Win32 aplikacích, tak i využití Windows Forms ve WPF. Pozornost si také zaslouží bezpečnost technologie WPF, ta zde sice není podrobně popsána, ale jako základ pro pochopení principu a možností je dostačující.

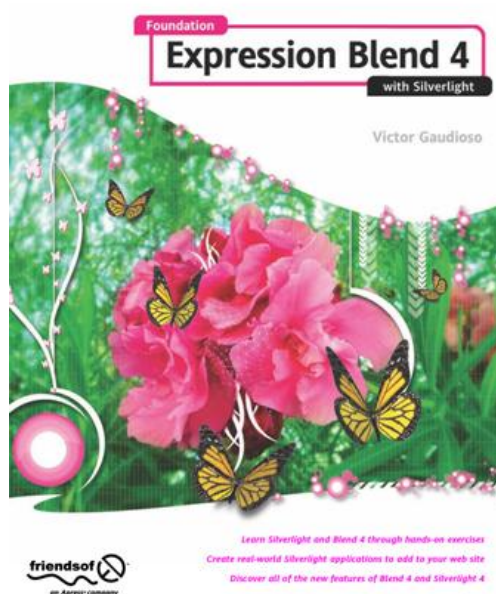
4.8 Pro WPF in VB 2010: Windows Presentation Foundation in .NET 4



Obr. 11. Pro WPF in VB 2010

Matthew MacDonald [20]. Kniha zaměřená na WPF v programovacím jazyku Visual Basic. Autor vysvětluje tematiku od začátku a předpokládá pouze znalost jazyka VB. V knize je popsána veškerá problematika velmi podrobně a rozsah knihy by měl stačit každému čtenáři, který se zajímá o tvorbu aplikací ve WPF. V jednotlivých kapitolách knihy autor vysvětlí základy práce s XAML a jeho propojení s VB, práci s dependency property, binding, tvorbu stylu, grafické chování komponent, animace, práci s videm a zvukem. V jedné z posledních kapitol se autor zmiňuje o možnosti provázání WPF a Windows Form, zde je důležitá skutečnost, že WPF může existovat spolu s Windows Form uvnitř jedné aplikace, ale navzájem musí zůstat odděleni.

4.9 Foundation Expression Blend 4 with Silverlight

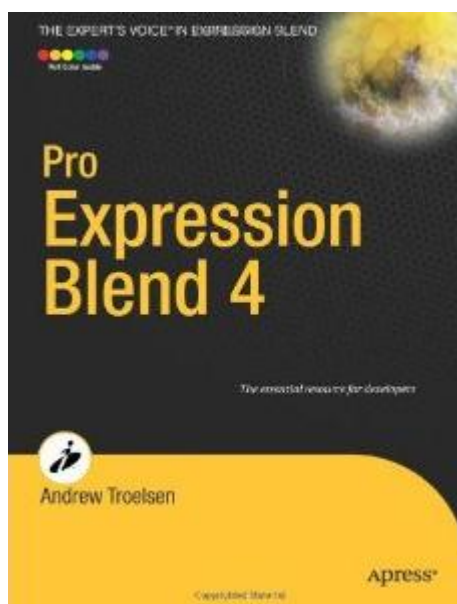


Obr. 12. Foundation Expression Blend 4

Victor Guadoso [6]. Na začátku knihy autor podrobně popisuje nastavení vývojového prostředí Expression Blend, které není na první pohled příliš jednoduché, hlavně pokud nemáte dřívější zkušenosti s jinými grafickými nástroji.

Kniha je zaměřená hlavně na Expression Blend a nejsou v ní podrobnější příklady stylování grafické vizualizace komponent, jako základ pro pochopení stylování a úpravy vzhledu prostřednictvím XAML je kniha vyhovující.

4.10 Pro Expression Blend 4



Obr. 13. Pro Expression Blend 4

Andrew Troelsen [9]. Kniha popisující Expression Blend a návod, jak s jeho pomocí vytvořit aplikace ve WPF nebo Silverlightu. Jsou zde popsány možnosti animace komponent, jejich stylování a práce s vektorovou grafikou. V knize najdete základní příklady stylování komponent, které stačí na pochopení principů vytváření vlastního vzhledu a chování ovládacích prvků.

4.11 Microsoft Developer Network Subscription Library (MSDN Library)



Obr. 14. MSDN

Microsoft [10]. Na tomto webu najdete technickou dokumentaci ke všem třídám a komponentám spolu s nejaktuálnějšími informacemi o nových technologiích společnosti Microsoft. Narozdíl od MSDN Subscriber je přístup bezplatný. Na stránce najdete příklady použití tříd a komponent, včetně popisu jejich vlastností a metod.

U tříd, které podporuje Silverlight nebo WPF je v příkladech použití komponenty uveden XAML kód, příklad použití v jazyce C# a jeho ekvivalent v jazyce VB.

XAML

```
<ComboBox Text="Is not open">
  <ComboBoxItem MouseMove="OnHover" Name="cbi1">Item1</ComboBoxItem>
  <ComboBoxItem MouseMove="OnHover" Name="cbi2">Item2</ComboBoxItem>
  <ComboBoxItem MouseMove="OnHover" Name="cbi3">Item3</ComboBoxItem>
</ComboBox>
```

Obr. 15. XAML

C#	VB
<pre>cbox = new ComboBox(); cbox.Background = Brushes.LightBlue; cboxitem = new ComboBoxItem(); cboxitem.Content = "Created with C#"; cbox.Items.Add(cboxitem); cboxitem2 = new ComboBoxItem(); cboxitem2.Content = "Item 2"; cbox.Items.Add(cboxitem2); cboxitem3 = new ComboBoxItem(); cboxitem3.Content = "Item 3"; cbox.Items.Add(cboxitem3); cv2.Children.Add(cbox);</pre>	<pre>Dim cbox As System.Windows.Controls.ComboBox Dim cboxitem As System.Windows.Controls.ComboBoxItem Dim cboxitem2 As System.Windows.Controls.ComboBoxItem Dim cboxitem3 As System.Windows.Controls.ComboBoxItem cbox = New ComboBox() cbox.Background = Brushes.LightBlue cboxitem = New ComboBoxItem() cboxitem.Content = "Created with Visual Basic." cbox.Items.Add(cboxitem) cboxitem2 = New ComboBoxItem() cboxitem2.Content = "Item 2" cbox.Items.Add(cboxitem2) cboxitem3 = New ComboBoxItem() cboxitem3.Content = "Item 3" cbox.Items.Add(cboxitem3) cv2.Children.Add(cbox)</pre>

Obr. 16. C# a VB

4.12 Stack Overflow



Obr. 17. Stack Overflow

Stack exchange, inc [11]. Služba funguje jako poradna. Kde je pro každý dotaz založen nový topik, ve kterém jsou zobrazovány odpovědi na položenou otázku. Zde je umožněno přehledné vkládání kódu včetně barevného formátování, což usnadňuje jeho čtení.

K odpovědím je možné hlasovat o jejich správnosti, vhodnosti a použitelnosti. Ta, která je uživateli ohodnocena jako nejlepší je přehledně označena jako nejlepší řešení. To

usnadňuje dalším čtenářům rychleji vyhledat řešení jejich problému, aniž by museli otázku pokládat znovu.

Tato služba je natolik rozšířená a populární, že na většinu dotazů v Googlu je odkaz na Stack Overflow umístěn mezi prvními.

4.13 The Code Project



Obr. 18. The Code Project

CodeProject [12]. Na stránce se setkává komunita vývojářů, která sdílí své znalosti, kódy, řešení problémů a další informace za účelem usnadnit ostatním programátorům jejich práci a urychlit řešení problémů.

Naleznete zde mnoho tutoriálů a článků z nejrůznějších programovacích jazyků včetně WPF a Silverlight. Je zde také sekce Quick Answer, která slouží jako poradna, ve které si návštěvníci navzájem radí a diskutují nad svými problémy.

5 SILVERLIGHT V PRAXI

Silverlightu se jako konkurentovi Flashe nepodařilo prosadit mezi běžné uživatele internetu, například na poli přehrávání videa, jako je tomu u serveru Youtube. Ten je v současné době nejnavštěvovanějším portálem pro přehrávání videí. Nicméně i Silverlight zaznamenal úspěch na poli online přehrávání a streamování videa v podobě zajištění online přenosu olympijských her ve Vancouveru. I přes to, že Silverlight není tak rozšířený jako Flash, je úspěšný na poli komerčních aplikací.

5.1 Silverlight na olympijských hrách

V roce 2008 na olympijských hrách v čínském Pekingu se Microsoft stal součástí této sportovní události jako zprostředkovatel online vysílání. Učinil tak ve spolupráci s NBC. Staral se o online přenosy v rozlišení až 720p. Stejně tak se prostřednictvím Silverlightu mohli dívat na streamované video fanoušci v roce 2010 při přenosech z XXI. olympijských her ve Vancouveru. [10]

Po oznámení, že i v roce 2010 se o online přenosy z olympijských her bude starat technologie společnosti Microsoft vyšel článek na serveru Živě:

Kromě moderních komunikačních kanálů je zajímavé sledovat, jaké internetové technologie budou média využívat k šíření obrazu a zpráv. Zde se opět podařil dobrý marketingový kousek Microsoftu, když se stejně jako u posledních letních her bude ke streamování sportovních přenosů používat jeho technologie Silverlight. Přímé přenosy bude opět vysílat zejména NBC na nbcolympics.com, kde se budou objevovat i záznamy, rozhovory a další zpravodajský materiál a to opět ve stejném formátu. NBC navíc jako hlavní vyhledávač používá Bing, rovněž od Microsoftu.

Silverlight poběží i na stránkách tuzemské České televize. Ta Olympijské hry ve Vancouveru pokryje samozřejmě v rámci televizní vysílání, ale živé přenosy plánuje vysílat i na svých webových stránkách. Tradiční formát Windows Media, který ČT používá pro své iVysílání a online archiv, nyní vystřídá další technologie Microsoftu. [13]

Na stránkách české televize je o online přenosech během XXI. zimní olympiády následující vyjádření:

Olympijský portál České televize zaznamenal během zimní olympiády v kanadském Vancouveru přes tři miliony návštěv. Současně videopřenosy na stránkách sledovalo až 62 000 lidí. [14]

Na konci roku 2010 vydal Microsoft na svých stránkách případovou studii s výsledky online přenosů ze zimních olympijských her:

NBC se spojilo s předními dodavateli technologií v čele s Microsoftem na pokrytí 4 485 hodin vysílání v HD kvalitě prostřednictvím internetové informační služby (IIS) Smooth Streaming, což je video přehrávač založený na Silverlightu. Kromě toho, že prostřednictvím této služby mohli sportovní fanoušci sledovat olympijské hry online, tak mnoho reklamních partnerů NBC zaznamenalo více než 15,8 milionů unikátních návštěv. Jednou z nejdůležitějších možností technologie Smooth Streaming a Silverlightu je poskytnutí nástroje pro prezentaci reklamy současně s přehráváním videa, v tomto případě přenosy z olympijských her. [15]



Obr. 19. Smooth Streaming

Tab. 1. Sledovanost ZOH 2010

2010 Zimní olympijské hry: fakta o online přenosech	
Počet odvysílaných akcí	300

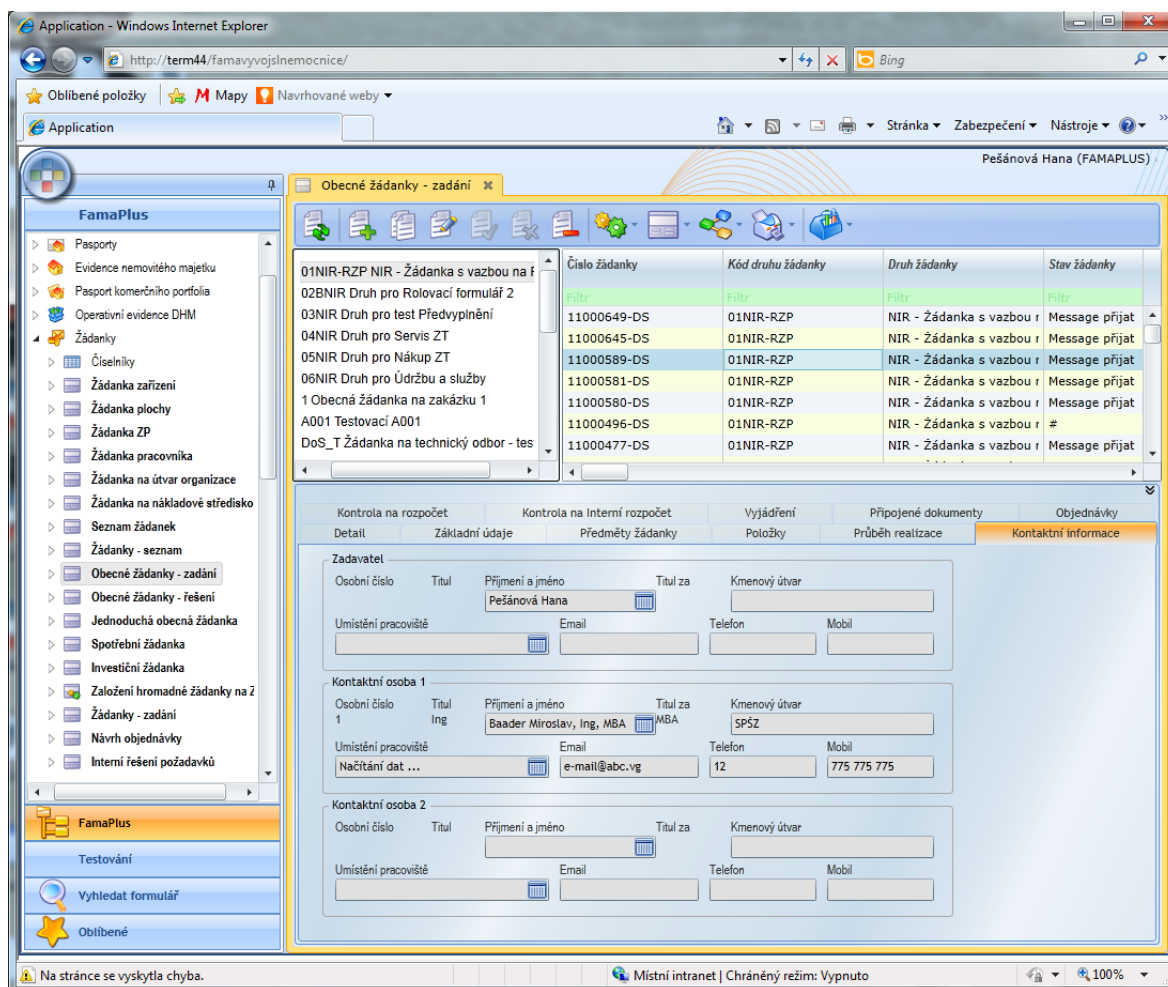
Počet odvysílaných hodin	4 400 000
Maximum současně vysílaných streamů	19
Celková velikost stažených videí	3,6 petabytů
Největší sledovanost v jednu momentu	181 000
Celkem odvysílaných videí	50 500 000
Počet unikátních diváků	15 800 000
Počet fotografií	41 560
Počet fotografických galerií	1 573
Kvalita videa	720p

5.2 Tesco SW a.s. - FaMa+

Informační systém FaMa+ je systém pro facility management (CAFM software), který pokrývá veškeré procesy evidence, provozu, správy a údržby movitého majetku i nemovitostí.

Software pro facility management FaMa+ je řešení modulární. Jednotlivé moduly je možné kombinovat tak, aby každá organizace získala CAFM software pouze pro ty procesy, nad kterými chce prostřednictvím systému mít kontrolu. Řešení se tak může skládat pouze z modulů bezprostředně souvisejících se správou majetku (např. Technický pasport či Evidence drobného hmotného majetku), nebo může obsahovat širší spektrum modulů pro správu veškerých podpůrných procesů firmy (např. Řízení externích vztahů, Žádankový systém (helpdesk), Doprava, Stěhování atd.) [16]

Software je založen na technologii Silverlight od společnosti Microsoft. Využívá širokých možností stylování, aby bylo možné pro jednotlivé aplikace vytvořit unikátní vzhled a přitom využívat již vytvořené funkcionality.



Obr. 20. TescoSW a.s. - Fama+

Historicky nejstarší vzhled má aplikace FaMa+, kde je v levé části umístěno hlavní menu, reprezentované stromovou strukturou, ze které se vybírají položky a po zvolení se otevře formulář skládající se ze seznamu. Ke každému záznamu se v dolní části vypisují podrobnosti. Nástupcem je potom vzhled aplikace MONIT, u které se hlavní menu přesunulo do horní části a je řešeno prostřednictvím menu button. Struktura formulářů zde byla zachována, pouze se změnil vzhled.

Další úpravou vzhledu vznikla aplikace Fama Services, zde se hlavní menu rozdělilo do několika částí (vlevo, nahoře a další podmenu opět v levé části).

Změn vzhledu v jednotlivých variantách bylo dosaženo pouze změnou XAML souborů, ve kterých je nadefinován vzhled aplikace. Na těchto příkladech je vidět jedna z výhod Silverlightu a tou je jednoduchá stylizace aplikace dle aktuálních potřeb, bez potřeby dalšího programování.

The screenshot displays the MONIT 2014 software interface. At the top, there's a user bar for 'Uživatel Isum (ISUM_USER)' with options to 'spravovat účet' and 'odhlásit'. Below this is a navigation menu with tabs: Program, Globální grant, Projekt, Číselníky, Přehledy, Správa aplikace, Integrační vazby, and Oblíbené. The main window is titled 'Program' and contains a sidebar with a 'Formuláře' menu listing various program-related forms. The central area shows a table of programs with columns: Číslo programu ČR, Název programu - český, Zkratka programu, Měna programu, Název stavu, and Zahájit. Below the table, there are tabs for 'Základní informace', 'Popis programu', and 'Harmonogram'. The 'Základní informace' tab is active, showing fields for program details like 'Zkratka programu' (IOP + OPT), 'Číslo programu ČR' (CZ.1.06), 'Číslo programu EK', 'Datum rozhodnutí EK', 'Název programu - český' (Program pro IOP), 'Název programu - anglicky' (Programme IOP), 'Měna programu' (CZK), 'Platí od' (1.1.2014), 'Kapitola SR' (317), 'Platí do' (31.12.2020), and a checked 'Víceúčelost' option. At the bottom, there are buttons for 'Realizace programu ŘO ukončena' and 'Stav Workflow'.

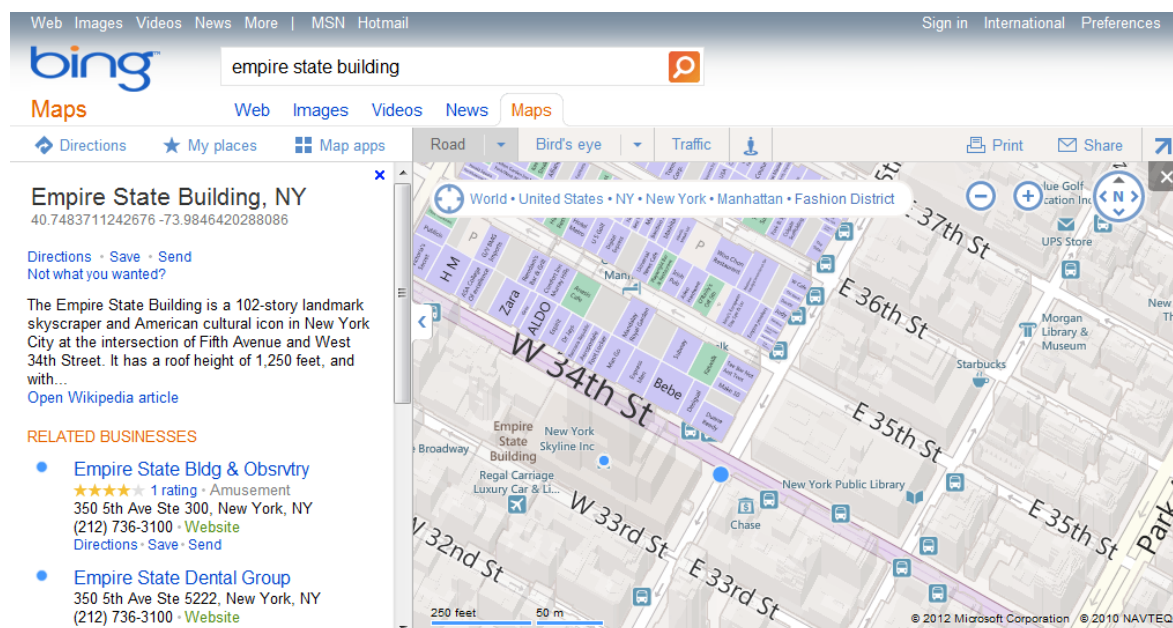
Obr. 21. TescoSW a.s. - Monit

The screenshot shows the FAMA GALATEK software interface. The top navigation bar includes icons for 'hlavní stránka', 'šablony', 'harmonogram', 'plánování', 'grafické výstupy', 'objekty', 'zaměstnanci', 'školení', 'zařízení a vybavení', 'revizní technici', and 'dodavatelé'. A sidebar on the left lists various equipment types under the heading 'REVIZE A KONTROLY'. The main content area displays a list of 'SYSTÉMOVÁ HLÁŠENÍ' (System Reports) with timestamps and descriptions of equipment checks. A chat window titled 'KOMUNIKACE' is open, showing a conversation between 'Adrijan' and 'GALATEK a.s.'. The chat messages include system status updates and error messages. At the bottom, a red banner indicates 'Máte aktivní 2 objednávky a 3 poptávky'.

Obr. 22. TescoSW a.s. - Fama Services

5.3 Bing Maps

V roce 2010 oznámil Microsoft, že základní verze Bing Maps poběží na technologii Silverlight. Po oznámení této novinky se během několika týdnů stala verze se Silverlight defaultní pro uživatele ze Spojených států. Stále zde ovšem byla možnost spustit Bing Maps v původní AJAX verzi a uživatelé tak měli možnost přepínat mezi těmito dvěma verzemi. [17]



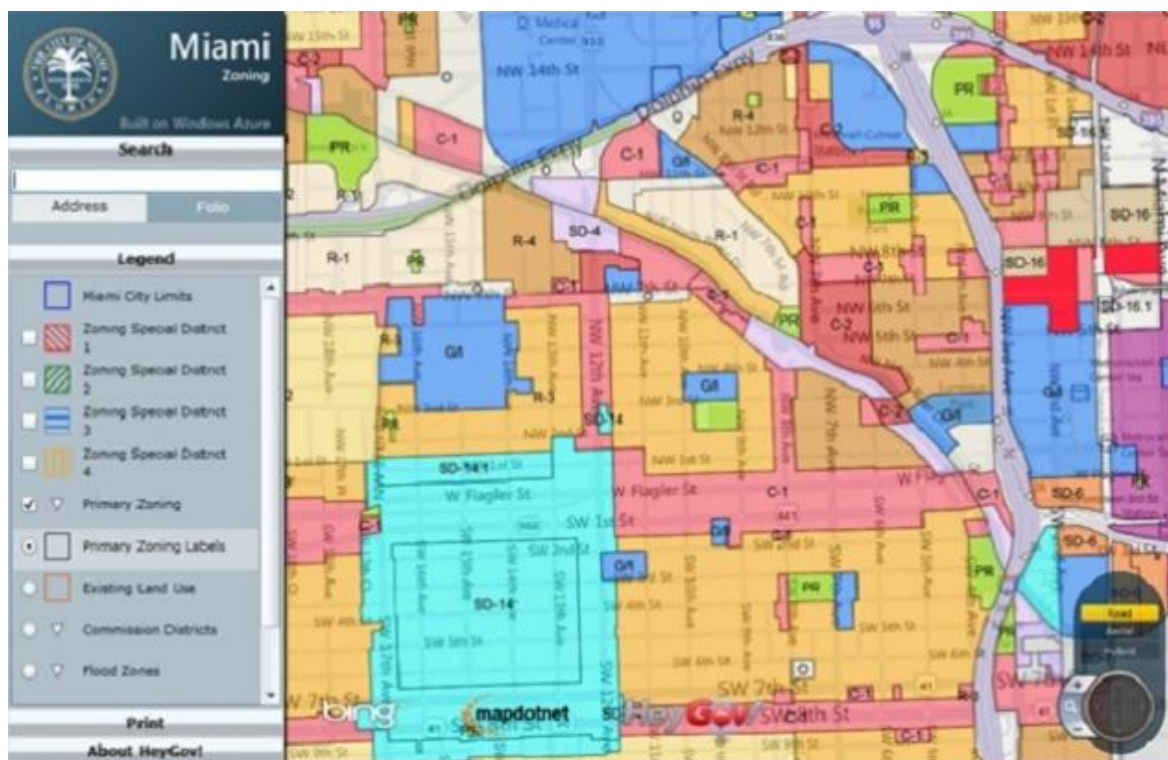
Obr. 23. Bind Maps

Přechod Bing Maps na Silverlight přinesl nové možnosti vývojářům, kterým se naskytla možnost využívat Bing Maps ve svých aplikacích.

Prostřednictvím Bing Maps Silverlight Controls spojující sílu Silverlightu a Bing Maps můžou vývojáři využít mocný nástroj pro práci s mapami. Vývojáři mohou využít Bing Maps Silverlight Control ve svých aplikacích při vyhledávání lokalit, míst a souřadnic.

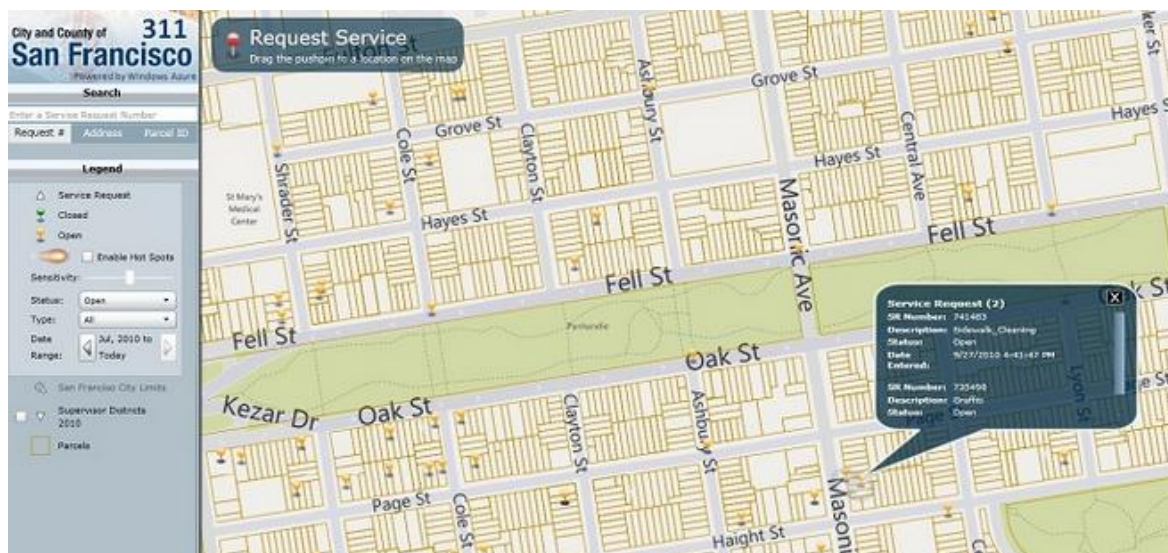
Bing Maps Silverlight Control Software Development Kit (SDK) obsahuje kompletní reference popisující celý interface, což usnadní jeho využití.[10]

V dubnu 2010 byla vydána aplikace HeyGov!, která byla navržena pro vládní účely a má usnadnit komunikaci s občany. Může ji využít kterékoliv vládní oddělení pro komunikaci s veřejností.



Obr. 24. HeyGov! - Miami

První města, která HeyGov! začala aktivně využívat, byla San Francisco a Miami. Aplikace je příkladem propojení Silverlightu a Bing Maps. [18]



Obr. 25. HeyGov! - San Francisco

6 WPF APLIKACE

Hlavní výhodou WPF před jeho předchůdcem Windows Forms je možnost, relativně jednoduše vytvořit bohaté uživatelské prostředí, které na první pohled upoutá svým originálním a bohatým obsahem. K tomu slouží hlavně značkovací jazyk XAML, který využívá k tvorbě uživatelského rozhraní a umožňuje od sebe oddělit funkčnost a vzhled aplikace.

6.1 Visual Studio 2010

Visual Studio je jeden z klíčových produktů firmy Microsoft a ve verzi 2010 bylo využito WPF k vytvoření jeho uživatelského prostředí.

WPF bylo navrženo k modernizování uživatelského prostředí a ke zvýšení jeho komfortu tím, že umožnilo vývojářům využít framework podporující DirectX 3D, animace, vektorovou grafiku, data binding, stylování a multimediální obsah. První verze WPF, podle mnoha vývojářů, poskytovala omezené nástroje k využití jeho potenciálu. [28]

Rockford Lhotka, hlavní technolog z Magenic Technologies Inc., při zveřejnění informace o tom, že Visual Studio 2010 bude využívat WPF řekl: "Je dobré vědět, že i Microsoft využívá WPF v jednom z jejich klíčových produktů. Budí to dojem, že WPF bude dlouhodobě podporováno a dojde k tvorbě nových nástrojů k jeho využívání. Pokaždé, když Microsoft používá své vlastní technologie je to ku prospěchu nás všech." [28]

6.2 Family.Show

Famaily.Show je určen pro nadšence zajímavící se o rodokmeny, mrtvé lidi a genealogii, která je až neuvěřitelně populární. Je to hlavně referenční aplikace, prezentující Windows Presentation Application. Byla vyvinuta pro Microsoft, aby ukázala možnosti WPF. [29]

Na stránkách aplikace je uvedena informace, podle které jsou v aplikaci použity všechny možnosti komponenty book - styly, resources, templates, data binding, animace, transformace pro prezentaci inovativního vzhledu klasického rodokmenu.

II. PRAKTICKÁ ČÁST

7 SILVERLIGHT VS FLASH VS JAVA

Pro porovnání použitelnosti a celkové využitelnosti Silverlightu jsem vybral konkurenční technologie z oblasti rich internet application JavaFX a Adobe Flash.

7.1 JavaFX



Obr. 26. JavaFX

JavaFX je vývojová platforma Java klienta, navržená pro jednoduchou tvorbu RIA aplikací, které vypadají a chovají se stejně, nezávisle na platformě. JavaFX je založen na technologii Java a nabízí bohatou sadu grafických a multimediálních komponent s podporou hardwarově akcelerované grafiky.

Na konferenci JavaOne, Sun Microsystems představil platformu JavaFX, která umožnila vývojářům tvorbu rich internet aplikací pro mobilní zařízení, osobní počítače, televize a ostatní uživatelská zařízení. [22]

7.2 Adobe Flash



Obr. 27. Adobe Flash

Platforma Adobe Flash je přední technologií pro tvorbu rich internet aplikací. Představuje integrovanou sadu technologií, obklopenou zavedeným ekosystémem podpůrných programů, obchodních partnerů a uživatelských komunit. Společně poskytují vše, co

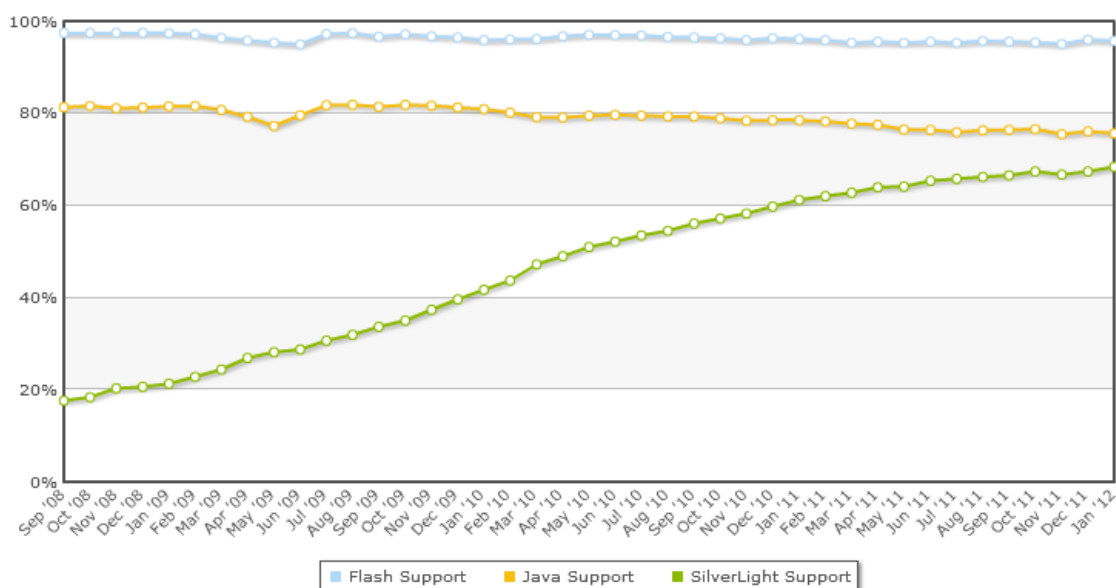
potřebujete pro vytváření a poskytování působivých aplikací, obsahu a videa co nejširšímu okruhu uživatelů. [23]

7.3 Využívání technologií pro RIA aplikace

Následující data jsou dostupná z statowl.com [24]. Jedná se o stránku zabývající se statistickou analýzou a průzkumem trhu v oblasti internetových trendů.

Tab. 2. Statistika podpory platforem u uživatelů

	Září 2008	Leden 2009	Leden 2010	Leden 2011	Leden 2012
Flash	97.48%	97.40%	95.89%	96.18%	95.74%
Java	81.37%	81.51%	80.95%	78.54%	75.63%
Silverlight	17.64%	21.31%	41.76%	61.19%	68.37%

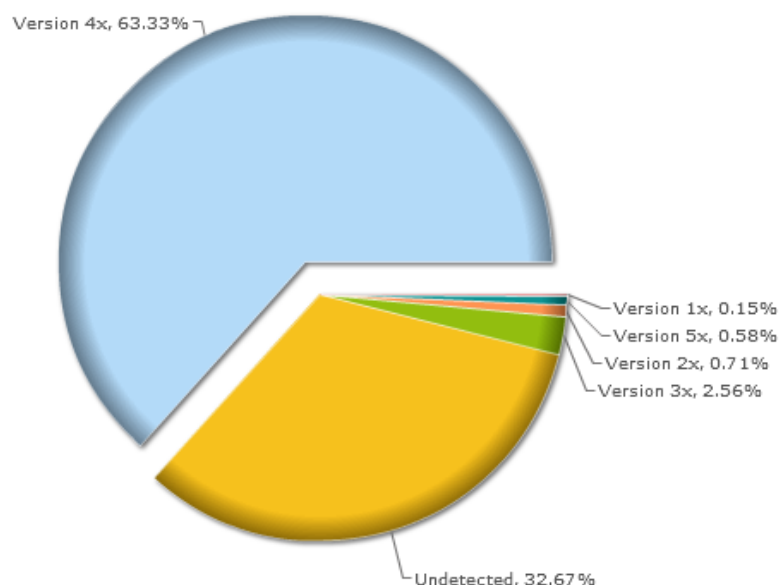


Obr. 28. Statistika podpory platforem u uživatelů

Z tabulky a z grafů lze vyčíst, že Silverlight postupně upevňuje svou pozici na trhu a za poslední tři roky stoupl počet počítačů, na kterých je nainstalován ze 17% na 68%. Sice stále nedosahuje stejného "pokrytí" jako Flash nebo Java, ale i tak lze říci, že v případě tvorby aplikace není počet uživatelů s nainstalovaným plug-inem překážkou a pokud bude pokračovat ve stejném trendu, měl být do roku 2013 rozšířenější platformou než Java.

Další graf zachycuje nainstalované verze Silverlightu. I když je verze Silverlight 5 vydána několik měsíců, je stále nejpoužívanější verze 4. Je to pravděpodobně z důvodu, že

jednotlivé aplikace zatím nepřešli na novější verzi a proto neměli uživatelé důvod ji instalovat.



Obr. 29. Nainstalované verze Silverlight

Pokud porovnáme počet pracovních pozic nabízených na Indeed.com, je vidět jednoznačná převaha Flashe, avšak oproti JavaFX je o Silverlight vývojáře podstatně větší zájem. Zde je ovšem otázkou, zda případní zaměstnavatelé do inzerátu neuvádějí pouze technologii Java.



Obr. 30. Nabídky práce pro internetové technologie

Pokud stejný graf budeme hledat na Google Trends, dostaneme výsledek, který je pravděpodobně skreslený dotazy laické veřejnosti (ta se může dotazovat na hry, videa atd.).



Obr. 31. Google Trends - Silverlight, Flash a JavaFX

Pro zajímavost ještě stejný graf bez Flashe:



Obr. 32. Google Trends bez Flashe

7.4 HTML 5

Úmyslně v předchozím textu není srovnání s HTML 5. Nejedná totiž o plug-in technologii a je zatím relativně nová. Také je otázkou, zda-li nebude trpět tím, že se každý prohlížeč při zobrazování HTML 5 bude chovat trochu jinak a tvůrci aplikací tak budou muset optimalizovat svou aplikaci pro jednotlivé prohlížeče zvlášť, což v případě Silverlightu a ostatních technologií, které potřebují nainstalovaný plug-in nehrozí.

8 PROČ WPF

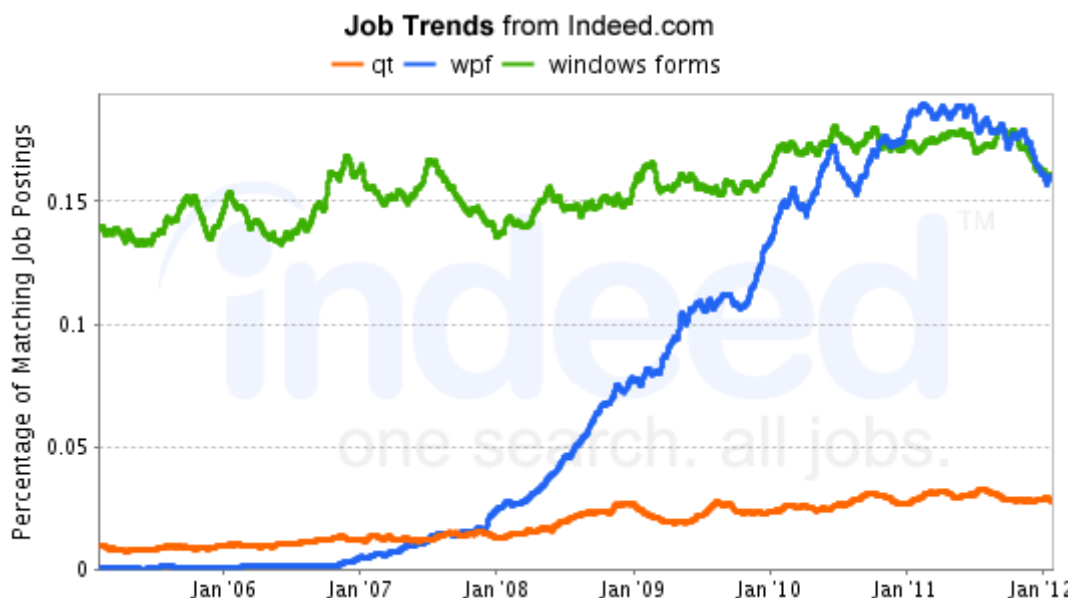
WPF je historickým nástupcem Windows Form, které se v dnešním světě mohou zdát jako zastaralé. A to hlavně z pohledu tvorby uživatelského rozhraní, kde mají omezené možnosti úpravy vzhledu komponent. Porovnávat WPF například z Javou není směřodonné, s ní by se dal srovnávat .Net. V následujícím textu bude porovnáváno Qt, WPF a Windows Forms.

8.1 Qt

Qt je multiplatformní framework pro vývojáře využívající jako programovací jazyk C++, QML, CSS a JavaScript. Jedná se o nástroj, který je vyvíjen jako Open Source a lze jej využívat na základě Open Source licence (LGPL v2.1) nebo podle obchodních podmínek v komerční sféře.

8.2 Rozšiřování WPF

Pokud můžeme něco považovat za směřodonné informace o rozšíření WPF v komerční sféře, tak jsou to pravděpodobně inzeráty na volná místa programátoru na této platformě.



Obr. 33. Job Trends - Qt, WPF, WinForms

Z grafu je vidět, že od počátků své existence, množství inzerátů na Indeed.com nabízející pracovní příležitost pro programátory na platformě WPF vyrovnalo zavedenému Windows

Forms a poslední dva roky se udržuje tento stav. V porovnání s Qt je zájem o WPF v komerční sféře podstatně větší.

Další možnost porovnat rozšířenost platformy nabízí Google Trends, který analyzuje položené dotazy v Googlu.



Obr. 34. Google Trends - Qt, WPF, WinForms

I zde je vidět podobný vývoj jako v případě Indeed.com. Dokonce zde WPF získalo jednoznačnou převahu již v roce 2007, zatímco v případě pracovních příležitostí to bylo až o tři roky později (ovšem s mnohem menším rozdílem). Z toho lze usuzovat, že vývojáři se o WPF začali zajímat mnohem dříve než se prosadil v komerční sféře.

8.3 WPF vs Windows Forms

WPF není přímo určen k tomu, aby nahradil Windows Forms. WinForms stále žije a jeho podpora ze strany Microsoftu i nadále pokračuje. WPF je pouze dalším nástrojem k tvorbě desktopových aplikací.

WPF není pouze o aplikacích, které "lahodí oku". Je sice pravda, že WPF nabízí širokou podporu pro tvorbu vizuálních efektů a animací, ale to není zdaleka vše, co nabízí. WPF je vhodné použít například pokud plánujete použít v aplikaci různé typy medií. Například video, dokumenty, 3D animace nebo různé přechody při prohlížení obrázků. Je také vhodné, pokud chcete vytvořit skinovatelné UI nebo když potřebujete pracovat s XML daty. [25]

Na druhou stranu Windows Form není definitivně ze hry i navzdory rychlému vývoji a rozšiřování WPF. Pokud vytváříte aplikace bez nutnosti rozsáhlé podpory funkcionality, kterou nabízí WPF, není důvod, proč opustit vyzkoušenou a osvědčenou platformu. [25]

Nakonec je zde jedna důležitá věc, možnost využití WPF komponent uvnitř Windows Forms a naopak, možnost použít Windows Form komponenty uvnitř WPF. Pokud tedy máte koupené nástroje třetích stran pro jednu platformu a potřebujete využít funkcionality platformy druhé, není nutné kvůli tomu měnit zavedenou platformu. [25]

Stejný závěr lze pravděpodobně uvést i v případě ostatních technologií. Pokud tedy nemusíte a nemáte k tomu vyloženě důvod, není nutné přecházet na technologii WPF, zvláště pokud máte nakoupené nástroje třetích stran, které by v případě přechodu nebyli použitelné. Pokud ale rozjíte nový projekt, u kterého dosud nemáte specifikovanou vývojovou platformu, stojí za zvážení využití právě WPF. Možnosti, které přináší mohou být velmi zajímavé a v budoucnu mohou ušetřit čas při úpravě vzhledu nebo při práci s multimediálním obsahem.

9 TESTOVACÍ OTÁZKY

Součástí diplomové práce je také aplikace s testovými otázkami pro ověření znalostí o platformě WPF a Silverlight. Aplikace je vytvořena ve WPF, otázky jsou uloženy v příloženém XML souboru, ze kterého jsou při spuštění testu načteny, deserializovány a uloženy do nachystané property.

9.1 Deserializace

Jak již bylo zmíněno, otázky jsou uloženy v XML formátu v příloženém souboru, který je načten a následně je z něj vytvořen seznam objektů, ve kterých jsou uloženy načtené informace.

```
<Question>
  <QuestionText>Který Layout umožňuje umisťovat
  komponenty zadáním absolutní hodnoty?</QuestionText>
  <ID>0</ID>
  <CorrectAnswere>b</CorrectAnswere>
  <Answers>
    <string>Grid</string>
    <string>Canvas</string>
    <string>WrapPanel</string>
    <string>DockPanel</string>
  </Answers>
</Question>
```

Obr. 35. Otázka v XML formátu po serializaci

```
public class Question : INotifyPropertyChanged
{
    public Question(){}
    public void QuestionChange(Question q){}

    string questionText;
    public string QuestionText{}

    int id;
    public int ID{}

    string correctAnswere;
    public string CorrectAnswere{}

    List<string> answers;
    public List<string> Answers{}

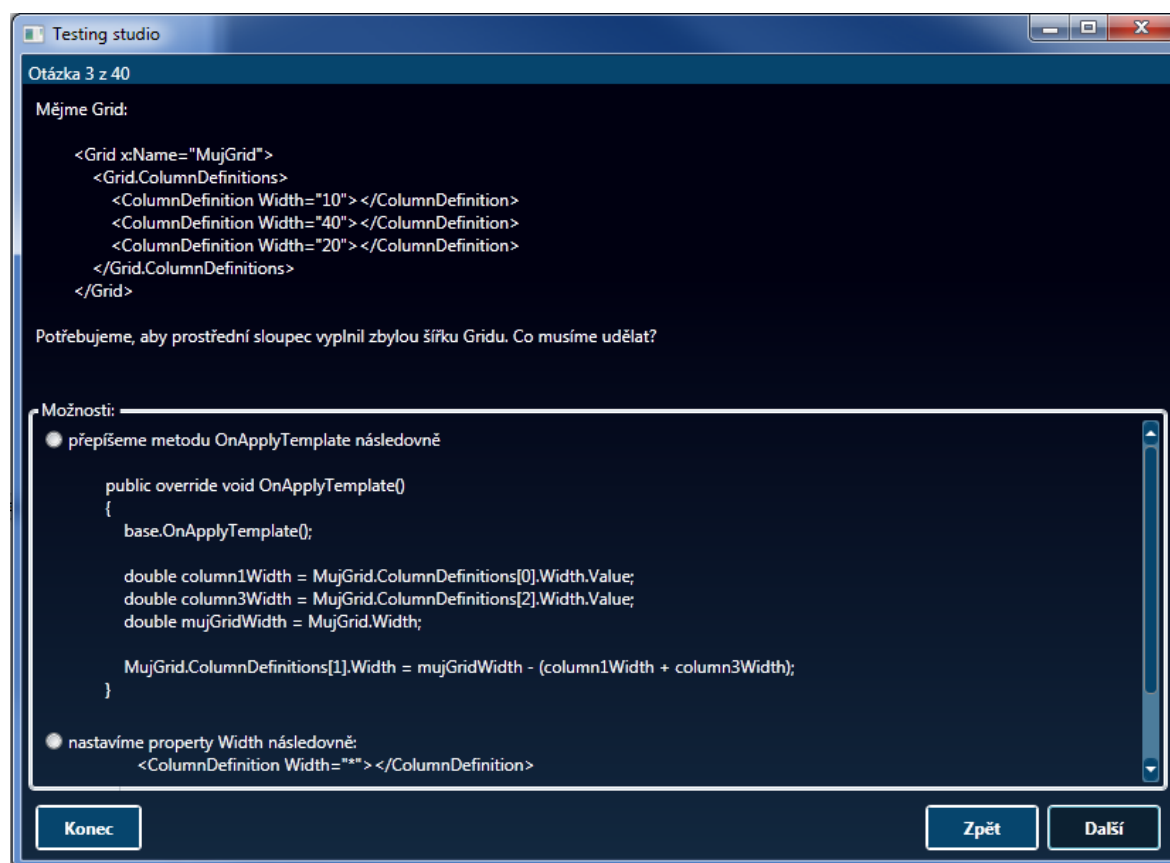
    string markedAnswere;
    public string MarkedAnswere{}

    public event PropertyChangedEventHandler PropertyChanged;
    protected void OnPropertyChanged(string name){}
}
```

Obr. 36. Struktura třídy

9.2 Testovací aplikace

Samotná aplikace je napsána ve WPF, vzhled aplikace je vytvořen pomocí Microsoft Expression Blend 4. Jednalo se převážně o stylování komponent Button a Scrollbar, u nichž je vygenerovaný původní styl a ten je následně upraven. Ostatní komponenty mají pouze nastavené základní property, jako je background nebo font.



Obr. 37. Aplikace pro testování znalostí

9.3 Ukázka testových otázek

Jako ukázka jsou přiloženy dvě otázky. Celý seznam čítá dohromady 40 otázek a je dostupný v přiložené aplikaci.

Otázka:

Máme komponentu TextBox. Jak nastavíme barvu písma na červenou?

Možnosti:

a) TextColor="Red"

b) `Foreground="Red"`

c) `Text.Color="Red"`

Správná odpověď je b.

Otázka:

Kterou z následujících hodnot nemůžeme nastavit do property `Visibility`?

Možnosti:

a) `Hidden`

b) `Collapsed`

c) `Visible`

d) `Show`

Správná odpověď je d.

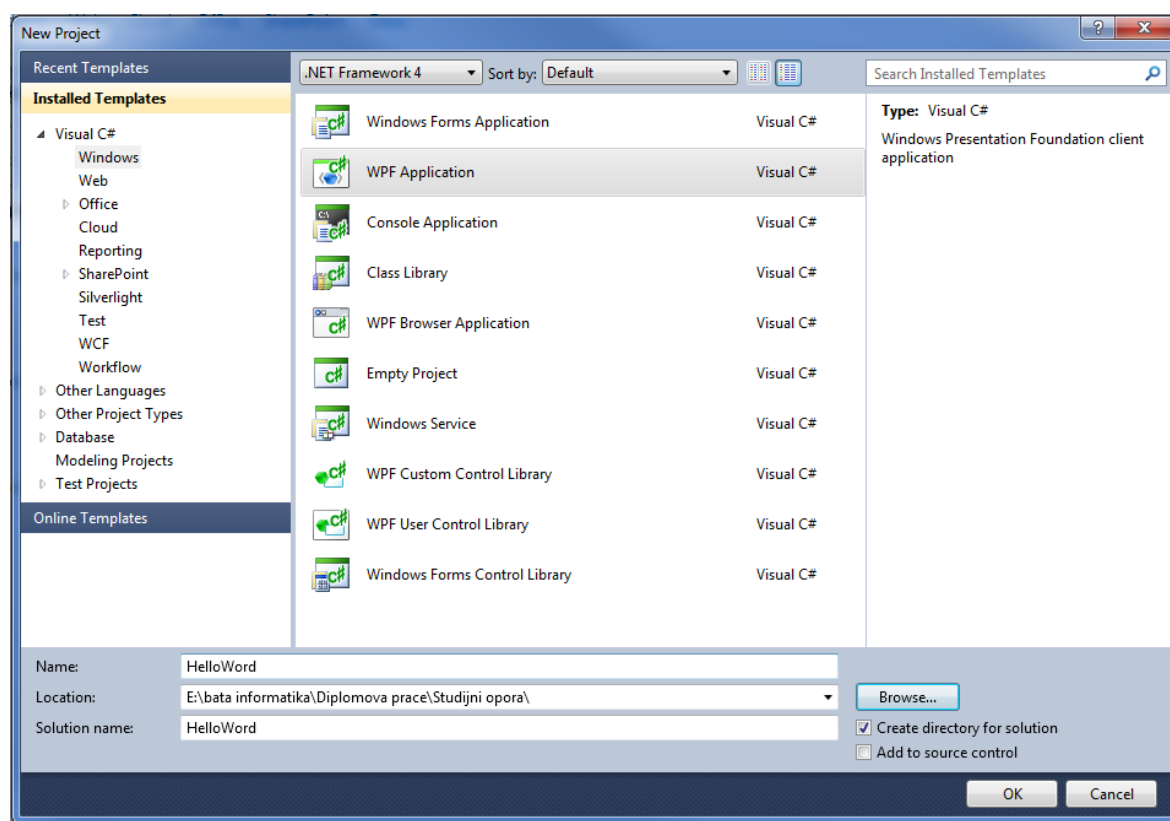
10 STUDIJNÍ OPORA

Následující kapitola má za úkol, provést čtenáře tvorbou první aplikace, vysvětlit základní pojmy a principy programování ve WPF a Silverlightu.

10.1 Hello Word

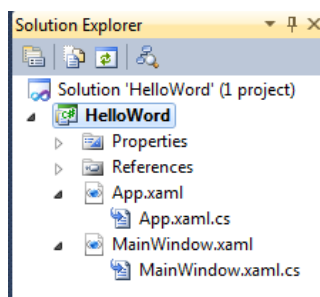
Začneme od samotného začátku a tím je vytvoření aplikace ve Visual Studiu.

Spustěte Visual Studio a vyberte "File", "New", "Project" v horním menu. V otevřeném okně vyberte WPF Application, zde zvolte kam se má aplikace uložit a pojmenujte ji.



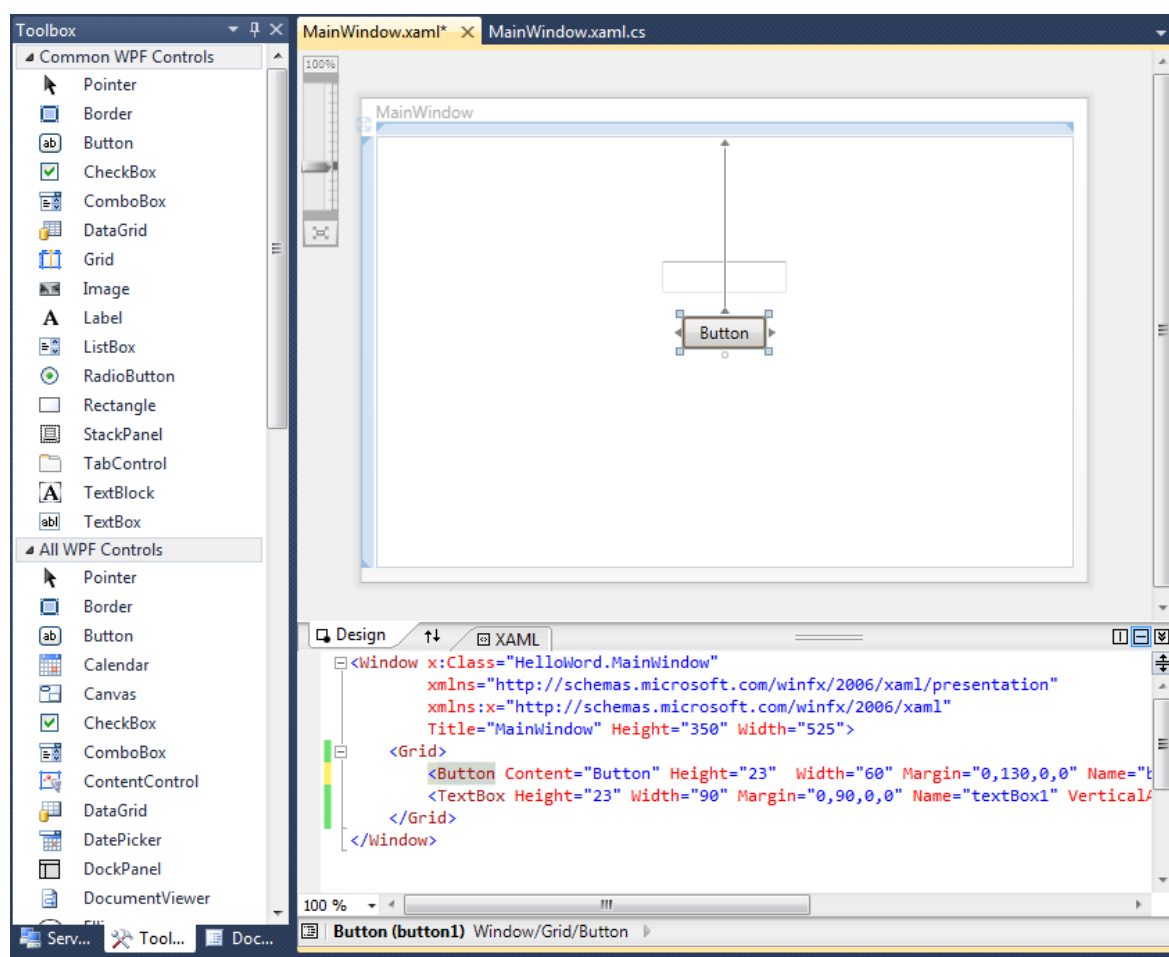
Obr. 38. Tvorba projektu ve Visual Studiu

Visual Studio automaticky vytvoří projekt spolu s několika soubory. App.xaml a MainWindow.xaml, spolu se soubory obsahující codebehind v jazyce, který je možné



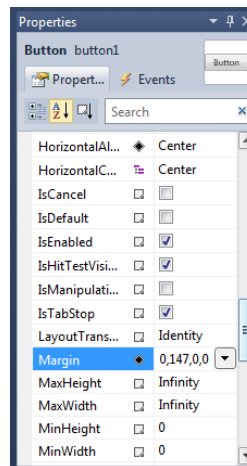
Obr. 39. Stuktura nově vytvořeného projektu

zvolit při vytváření nového projektu, v našem případě se jedná o C#. Otevřete soubor MainWindow.xaml a v editoru do hlavního okna umístěte Button a TextBox, jak je vidět na následujícím obrázku.



Obr. 40. Tvorba grafického rozhraní

V okně Properties nastavte tlačítku property HorizontalAlignment na center, VerticalAlignment na top a property Margin=0,150,0,0. Tím docílíme toho, že tlačítko bude vždy 150 bodů od horní části aplikace a podle vodorovné osy bude ve středu.



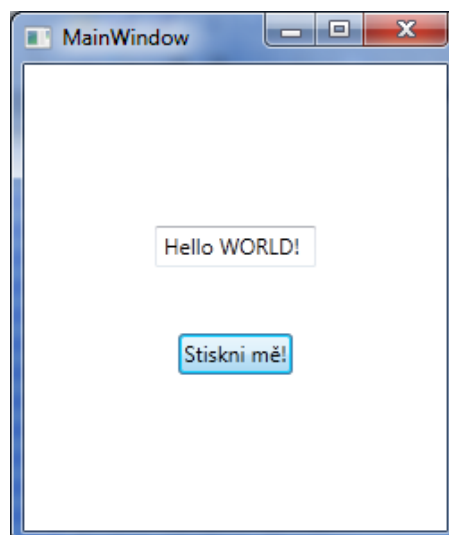
Obr. 41. Nastavování vlastností

Do property Content napište "Stiskni mě!". Dále v záložce Events doubleclickem vpravo od události Click vytvoříme metodu v codebehind. Do metody, kterou Visual Studio vygenerovalo automaticky přidejte následující kód:

```
textBox1.Text = "Hello WORLD!";
```

TextBox nastavte property HorizontalAlignment a VerticalAlignment stejně jako v případě komponenty Button a Margin nastavte na hodnotu 0,90,0,0.

Aplikaci spustíte z horního menu "Debug" a "Start Debugging" nebo stisknutím klávesy F5.



Obr. 42. Výsledná aplikace Hello World

10.2 Studijní opora

Nyní již vytvoříme aplikaci, která bude něco umět. Aplikace bude sloužit k zadání údajů o osobě, bude je umět uložit a zpětně načíst. Pro její tvorbu využijeme binding, stylování, serializaci a deserializaci.

Nejprve vytvořte nový projekt, stejně jako v předchozím případě.

10.2.1 Vytvoření třídy Osoba

V Solution Exploreru klikneme pravým tlačítkem myši na projekt a zvolíme "Add", "New Item...", v otevřeném okně zvolte Class a soubor pojmenujte Osoba.cs. Za definici třídy přidejte interface INotifyPropertyChanged, ze kterého bude třída Osoba dědit. Budete muset přidat jmenný prostor System.ComponentModel a implementovat tento interface přidáním property: public event PropertyChangedEventHandler PropertyChanged.

Rozhraní INotifyPropertyChanged slouží k tomu, aby jeden objekt mohl sledovat druhý. Sledující objekt je informován o každé změně ve sledovaném objektu. Informaci o změně property dostanou prostřednictvím události PropertyChanged. Klient tedy může reagovat na nová data. Přidáme ještě metodu, která usnadní využívání PropertyChanged a pojmenujeme si ji OnPropertyChanged s jedním parametrem typu string. Stav třídy Osoba je tedy následující:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.ComponentModel;

namespace WPFopora
{
    class Osoba : INotifyPropertyChanged
    {
        public event PropertyChangedEventHandler PropertyChanged;

        protected void OnPropertyChanged(string name)
        {
            PropertyChangedEventHandler handler = PropertyChanged;
            if (handler != null)
            {
                handler(this, new PropertyChangedEventArgs(name));
            }
        }
    }
}
```

Do třídy přidáme proměnné jmeno, prijmeni, mesto, ulice, pohlavi. Tyto budou typu string. Dále proměnnou typu int smerovacíCislo a typu DateTime datumNarozeni. Aby jsme mohli využít binding a rozhraní INotifyPropertyChanged přidáme ke každé proměnné stejně pojmenovanou property, která bude začínat velkým písmenem. Část get bude vracet hodnotu z odpovídající proměnné, v části set tuto proměnnou nastavíme a zavoláme metodu OnPropertyChanged s parametrem nastaveným na jméno property. Tedy následovně:

```
private string jmeno;
public string Jmeno
{
    get
    {
        return jmeno;
    }
    set
    {
        jmeno = value;
        OnPropertyChanged("Jmeno");
    }
}

private string prijmeni;
public string Prijmeni
{
    get
    {
        return prijmeni;
    }
    set
    {
        prijmeni = value;
        OnPropertyChanged("Prijmeni");
    }
}

private string mesto;
public string Mesto
{
    get
    {
        return mesto;
    }
    set
    {
        mesto = value;
        OnPropertyChanged("Mesto");
    }
}

private string ulice;
public string Ulice
{
    get
    {
        return ulice;
    }
}
```

```
    }
    set
    {
        ulice = value;
        OnPropertyChanged("Ulice");
    }
}

private int smerovaciCislo;
public int SmerovaciCislo
{
    get
    {
        return smerovaciCislo;
    }
    set
    {
        smerovaciCislo = value;
        OnPropertyChanged("SmerovaciCislo");
    }
}

private DateTime datumNarozeni;
public DateTime DatumNarozeni
{
    get
    {
        return datumNarozeni;
    }
    set
    {
        datumNarozeni = value;
        OnPropertyChanged("DatumNarozeni");
    }
}

private string pohlavi;
public string Pohlavi
{
    get
    {
        return pohlavi;
    }
    set
    {
        pohlavi = value;
        OnPropertyChanged("Pohlavi");
    }
}
```

Tímto dosáhneme toho, že v případě nastavení kterékoliv property bude klient o této změně informován.

10.2.2 Tvorba grafického rozhraní

Nyní otevřeme soubor MainWindow.xaml a umístíme na něj ovládací prvky, které umožní nastavit třídu Osoba. Na obrázku, můžete vidět jedno z možných rozvržení.

Obr. 43. Rozvržení UI aplikace

Jméno, příjmení, Město, Ulice a Směrovací číslo se budou zadávat prostřednictvím komponenty `TextBox`. K zadání data narození použijeme `DatePicker` a pohlaví se bude vybírat z `ComboBoxu`. Popisky jsou tvořeny komponentami `TextBlock` a ohraničení okolo města, ulice a směrovacího čísla je realizováno pomocí `GroupBoxu`.

10.2.3 Codebehind MainWindow

Nejdříve v `MainWindow.xaml` pojmenujeme `ComboBox` pro zadávání pohlaví nastavením property `Name="cbPohlavi"` a v konstruktoru `MainWindow` do něj přidáme prvky Muž, Žena, Neuvedeno a hodnotu Neuvedeno nastavíme, jako `selected`.

Do `MainWindow` přidáme proměnnou a property, ve které bude uložen objekt `Osoba` a v konstruktoru `MainWindow` zavoláme konstruktor třídy `Osoba` a v objektu nastavíme `DatumNarozeni` na aktuální datum. Stav `MainWindow`:

```
public partial class MainWindow : Window
{
    public MainWindow()
    {
        InitializeComponent();
        AktOsoba = new Osoba();

        AktOsoba.DatumNarozeni = DateTime.Today;
        cbPohlavi.Items.Add("Muž");
        cbPohlavi.Items.Add("Žena");
        cbPohlavi.Items.Add("Neuvedeno");
        cbPohlavi.SelectedItem = "Neuvedeno";
    }
}
```

```

private Osoba osoba;
public Osoba AktOsoba
{
    get { return osoba; }
    set
    {
        osoba = value;
    }
}
}

```

10.2.4 Binding

Vrátíme se do MainWindow.xaml, kde pojmenujeme jednotlivé komponenty pro zadávání hodnot. (např.: tbJmeno, tbPrijmeni, tbMesto, tbUlice, tbSmerovaciCislo, dpNarozeni, cbPohlavi). Vytvoříme binding na příslušné property objektu Osoba. Jeden z možných způsobů bindingu:

```

<TextBox Height="23" HorizontalAlignment="Left" Margin="45,6,0,0" Name="tbMesto"
Text="{Binding Mesto, Mode=TwoWay}" VerticalAlignment="Top" Width="120" />

```

```

<TextBox Height="23" HorizontalAlignment="Left" Margin="215,6,0,0" Name="tbUlice"
Text="{Binding Ulice, Mode=TwoWay}" VerticalAlignment="Top" Width="120" />

```

```

<TextBox Height="23" HorizontalAlignment="Left" Margin="6,60,0,0"
Name="tbSmerovaciCislo" Text="{Binding SmerovaciCislo, Mode=TwoWay}"
VerticalAlignment="Top" Width="120"/>

```

```

<TextBox Height="23" Margin="102,9,0,0" Name="tbJmeno" Text="{Binding Path=Jmeno,
Mode=TwoWay}" VerticalAlignment="Top" HorizontalAlignment="Left" Width="120" />
<TextBox Height="23" Margin="282,12,0,0" Name="tbPrijmeni" VerticalAlignment="Top"
Text="{Binding Prijmeni, Mode=TwoWay}" HorizontalAlignment="Left" Width="120" />

```

```

<DatePicker Height="25" HorizontalAlignment="Left" Margin="102,39,0,0"
Name="dpNarozeni" SelectedDate="{Binding DatumNarozeni, Mode=TwoWay}"
VerticalAlignment="Top" Width="120" />

```

```

<ComboBox Height="23" HorizontalAlignment="Left" Margin="282,41,0,0" Name="cbPohlavi"
SelectedValue="{Binding Pohlavi, Mode=TwoWay}" VerticalAlignment="Top" Width="120" />

```

Tímto jsme nastavili property na grafické komponentě, která bude provázána s property objektu. Zatím ovšem není nastaven objekt, ze které se bere navázaná property. Ten nastavíme v konstruktoru MainWindow.xaml.cs následovně:

```

public MainWindow()
{
    InitializeComponent();
    AktOsoba = new Osoba();

    tbJmeno.DataContext = AktOsoba;
    tbMesto.DataContext = AktOsoba;
    tbPrijmeni.DataContext = AktOsoba;
    tbSmerovaciCislo.DataContext = AktOsoba;
}

```

```
tbUlice.DataContext = AktOsoba;  
cbPohlavi.DataContext = AktOsoba;  
dpNarozeni.DataContext = AktOsoba;  
  
AktOsoba.DatumNarozeni = DateTime.Today;  
cbPohlavi.Items.Add("Muž");  
cbPohlavi.Items.Add("Žena");  
cbPohlavi.Items.Add("Neuvedeno");  
cbPohlavi.SelectedItem = "Neuvedeno";  
}
```

Další možnost jak realizovat binding je, že zápis v codebehind, místo v xaml souboru. Pak bychom ho napsali následujícím způsobem:

```
tbJmeno.SetBinding(TextBox.TextProperty, new Binding("Jmeno") { Mode =  
BindingMode.TwoWay });
```

Všimněte si, že u bindingu je nastaven Mode na TwoWay, to znamená, že binding funguje oboustraně, z GUI na property objektu a z property zpět do GUI. Vyzkoušet si to můžete například přidáním tlačítka, které nastaví některou z vlastností objektu a toto nastavení se promítne zpět do GUI, v opačném směru například přidáním MessageShow do set části některé property objektu Osoba.

Další možnosti nastavení jsou:

- TwoWay - aktualizuje cílovou property, kdykoliv se změní zdrojová property a obráceně
- OneWay - aktualizuje cílovou property, kdykoliv se změní zdrojová property
- OneTime - aktualizuje cílovou property, pouze při startu aplikace, nebo při nastavení DataContext
- OneWayToSource - aktualizuje zdrojovou property, když se změní property cílová
- Default - použije se výchozí nastavení cílové property

(pozn.: cílová property je ta, u které nastavujeme binding a zdrojová je ta, na kterou se odkazujeme. V předchozím příkladu je cílová property TextProperty na TextBoxu a zdrojová je property Mesto, objektu Osoba.)

10.2.5 Serializace a Deserializace

Serializace je v podstatě převedení datového typu na stream, který může být uložen do nějakého perzistentního úložiště (soubor, databáze). Deserializace je potom opačný proces.

Nejprve přidáme atribut [Serializable], před definici třídy.

```
[Serializable]
public class Osoba : INotifyPropertyChanged
```

Zde je potřeba si uvědomit, že jsou serializovány všechny datové členy dané třídy. Je důležité, aby všechny tyto členy byly samy o sobě serializovatelné, což naše třída Osoba splňuje. Pokud by však obsahovala datový typ, který není serializovatelný, aplikace by vyvolala výjimku System.Runtime.Serialization.SerializationException. To by například mohlo nastat, pokud bychom bydlíště realizovali jako samostatnou třídu a tu nenastavili jako Serializable.

Nyní do grafického rozhraní přidáme dvě tlačítka Load a Save včetně metod reagujících na kliknutí, stejně jako jsme přidávali tuto metodu v příkladě Hello World. V metodě Save nejprve serializujeme objekt Osoba a poté ho uložíme do souboru. V reakci na stisknutí Load tento soubor otevřeme, načteme z něj uložený objekt, který deserializujeme.

10.2.5.1 Serializace

Serializovaný objekt budeme ukládat ve formátu xml do souboru, který si uživatel sám vybere. Proto musíme nejprve otevřít SaveFileDialog pro výběr souboru. Poté vytvoříme serializer a aktuální objekt uložíme do souboru.

```
private void btnSave_Click(object sender, RoutedEventArgs e)
{
    Microsoft.Win32.SaveFileDialog dlg = new Microsoft.Win32.SaveFileDialog();
    dlg.DefaultExt = ".xml";
    dlg.Filter = "Text documents (*.xml)|*.xml";

    Nullable<bool> result = dlg.ShowDialog();

    string filename = "";
    if (result == true)
    {
        filename = dlg.FileName;
    }
    else
        return;

    StreamWriter textWriter = new StreamWriter(filename);
    XmlSerializer Xml_ser = new XmlSerializer(typeof(Osoba));
    Xml_ser.Serialize(textWriter, osoba);
    textWriter.Close();
}
```

Pokud nyní spustíte aplikaci, vyplníte údaje a objekt uložíte, výsledek by měl vypadat následovně:

Obr. 44. Aplikace s vyplněnými údaji

```
<Osoba xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <Jmeno>Petr</Jmeno>
  <Prijsmeni>Novák</Prijsmeni>
  <Mesto>Praha</Mesto>
  <Ulice>Letenská</Ulice>
  <SmerovaciCislo>11800</SmerovaciCislo>
  <DatumNarozeni>1970-07-17T00:00:00</DatumNarozeni>
  <Pohlavi>Muž</Pohlavi>
</Osoba>
```

10.2.5.2 Deserializace

Nyní načteme již uložená data a zobrazíme je v naší aplikaci. Jedná se tedy o opačný proces než při serializaci. Musíme nejprve vybrat uložený soubor, deserializovat data uložená v něm a nakonec tato data uložit do objektu typu Osoba.

Nejprve tedy otevřeme dialogové okno OpenFileDialog a nastavíme filtr na soubory s koncovkou xml. Poté, co uživatel soubor vybere, otevřeme soubor pomocí XmlTextReader, data deserializujeme a současně je uložíme do nové instance objektu Osoba. Nyní stačí nastavit vlastnosti objektu, která je nabídnována na grafické rozhraní aplikace.

```
private void btn_Load_Click(object sender, RoutedEventArgs e)
{
    Microsoft.Win32.OpenFileDialog dlg = new Microsoft.Win32.OpenFileDialog();

    dlg.DefaultExt = ".xml";
```

```

dlg.Filter = "Text documents (*.xml)|*.xml";

Nullable<bool> result = dlg.ShowDialog();

string filename = "";
if (result == true)
{
    filename = dlg.FileName;
}
else
    return;

TextReader textReader = new StreamReader(filename);
XmlSerializer Xml_Serializer = new XmlSerializer(typeof(Osoba));
XmlTextReader XmlReader = new XmlTextReader(textReader);

try
{
    Osoba pomOsoba = Xml_Serializer.Deserialize(XmlReader) as Osoba;
    AktOsoba.Jmeno = pomOsoba.Jmeno;
    AktOsoba.Prijmeni = pomOsoba.Prijmeni;
    AktOsoba.Pohlavi = pomOsoba.Pohlavi;
    AktOsoba.Mesto = pomOsoba.Mesto;
    AktOsoba.DatumNarozeni = pomOsoba.DatumNarozeni;
    AktOsoba.SmerovaciCislo = pomOsoba.SmerovaciCislo;
    AktOsoba.Ulice = pomOsoba.Ulice;
}
catch
{
    MessageBox.Show("Soubor se nepodarilo nacist");
}
textReader.Close();
}

```

Deserializace objektu je uzavřena v try bloku pro případ, že by uživatel vybral soubor, který neobsahuje serializovaný objekt typu Osoba. Pokud se podíváte na průběh deserializace zjistíte, že pro nastavení AktOsoba je potřeba poměrně dlouhý kód. Je to z toho důvodu, že jsme nabíndovali grafické prvky přímo na property objektu AktOsoba. Pokud bychom tedy pouze přiřadili AktOsoba = pomOsoba, nedojde k zavolání OnPropertyChanged uvnitř třídy Osoba na jednotlivých vlastnostech. Proto nyní změníme binding pro zjednodušení deserializace.

10.2.6 Binding 2

Nejprve upravíme třídu Osoba, odstraníme rozhraní INotifyPropertyChanged, ze kterého třída dědí a spolu s ním i veškerý kód, který s tím souvisí. To znamená metodu OnPropertyChanged a její volání v set částí všech property, nakonec i samotnou událost PropertyChanged.

Přesuneme reakci na změnu property o úroveň výš a implementujeme rozhraní `INotifyPropertyChanged` ve třídě `MainWindow`, stejně jako jsme to udělali již dříve ve třídě `Osoba`. To zahrnuje přidat interface `INotifyPropertyChanged`, ze kterého bude `MainWindow` dědit, přidat do `using System.ComponentModel` a událost `PropertyChanged` spolu s metodou `OnPropertyChanged`, kterou budeme volat v set části property `AktOsoba`.

Nyní upravíme binding komponent v grafickém rozhraní aplikace. Abychom nemuseli nastavovat `DataContext` každé komponentě zvlášť, pojmenujeme uvnitř `MainWindow.xaml` kořenovou komponentu `window` nastavením property `x:Name="_this"` a hlavnímu `Gridu`, ve kterém jsou umístěny ostatní prvky nastavíme `DataContext={Binding ElementName=_this}`. Tím nastavíme výchozí `DataContext` všem prvkům umístěným v tomto `gridu`. Pokud bychom chtěli, aby některá komponenta měla jiný `DataContext`, stačí ho v její definici změnit. Posledním krokem je změna bindingu na jednotlivých komponentách následovně:

```
Text="{Binding AktOsoba.Jmeno, Mode=TwoWay}"
```

Tímto jsme zjednodušili binding, který je nyní celý v souboru `xaml` a v `codebehind` s ním vůbec nepracujeme. Nakonec ještě upravíme deserializaci objektu tak, že deserializovaný objekt přímo přiřadíme do property `AktOsoba`:

```
AktOsoba = Xml_Serializer.Deserialize(XmlReader) as Osoba;
```

10.2.7 Stylování

Aplikace nyní umí vše, co je potřeba pro práci s objektem a schází už jen upravit její vzhled. Nejprve nastavíme pozadí hlavního `Gridu` na přechod z černé na tmavě modrou:

```
<Grid DataContext="{Binding ElementName=_this}">
    <Grid.Background>
        <LinearGradientBrush EndPoint="0.5,1" StartPoint="0.5,0">
            <GradientStop Color="Black" Offset="0.273" />
            <GradientStop Color="#FF12283F" Offset="1" />
        </LinearGradientBrush>
    </Grid.Background>
    ...
    ...
</Grid>
```

Tím se staly veškeré popisky v TextBlock komponentách nečitelné, protože mají černou barvu textu. Abychom nemuseli nastavovat property Foreground u všech komponent zvlášť vytvoříme defaultní styl, který jim barvu textu automaticky nastaví na bílou.

Nejprve přidáme do MainWindow.xaml sekci Window.Resources a v ní vytvoříme první styl pro TextBlock, který pouze nastaví barvu textu na bílou.

```
<Window x:Class="WPFopora.MainWindow"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        Title="MainWindow" Height="286" Width="443" x:Name="_this">
    <Window.Resources>
        <Style TargetType="TextBlock">
            <Setter Property="TextWrapping" Value="NoWrap"/>
            <Setter Property="TextTrimming" Value="None"/>
            <Setter Property="Foreground" Value="White" />
        </Style>
    </Window.Resources>
    <Grid DataContext="{Binding ElementName=_this}">
        ...
        ...
    </Grid>
</Windows>
```

Pokud bychom chtěli, aby nový styl využívaly pouze některé komponenty TextBlock, stačilo by vytvořit styl následovně:

```
<Style x:Key="TextBlockStyle1" TargetType="{x:Type TextBlock}">
    <Setter Property="TextWrapping" Value="NoWrap"/>
    <Setter Property="TextTrimming" Value="None"/>
    <Setter Property="Foreground" Value="White" />
</Style>
```

A pro použití pro konkrétní TextBlock se musí nastavit property Style:

```
<TextBlock Height="23" HorizontalAlignment="Left" Margin="12,41,0,0"
Name="textBlock6" Text="Datum narození" VerticalAlignment="Top"
Style="{DynamicResource TextBlockStyle1}" />
```

Dalším nečitelným popiskem je hlavička komponenty GroupBox, ale protože je použita pouze jednou, je jednodušší nastavit vlastnost Foreground="White".

Tímto je vytvořena aplikace, která umí nastavovat vlastnosti objektu, uložit je to xml souboru, opětovně je načíst a její vzhled je upraven, jak nastavením property tak i stylováním.

Pokud budete stylovat jednotlivé komponenty je nejjednodušší využít Expression Blend 4 a nechat si vygenerovat již existující styl komponenty a ten dále upravovat. Tím převezmáte

již existující funkcionalitu, kterou poté nemusíte pracně tvořit. To uděláte jednoduše tak, že kliknete pravým tlačítkem myši na komponentu, kterou chcete stylovat, zvolíte Edit Template a nakonec zvolíte Edit a Copy. Expression Blend vám vygeneruje současný styl včetně veškeré funkcionality. Pokud nemáte k dispozici Expression Blend, jsou základní styly k dispozici na MSDN a nejjednodušší způsob jak je dohledat, je zadat do Googlu například dotaz: style and template TextBox. Stylování komponent přímo ve Visual Studiu je poněkud komplikovanější, protože neobsahuje dostatečně vybavený editor stylů. K vytváření a úpravě grafických vlastností aplikace je proto vhodnější použít Expression Blend, který je k tomu přímo určen a nabízí mocnější nástroje na úpravu vzhledu a stylování.

ZÁVĚR

Diplomová práce obsahuje rešerši literárních zdrojů, dostatečně pokrývající informace o technologiích WPF a Silverlight. Uvedené zdroje jsou jak v tištěném knižním formátu, tak i v elektronické formě. V rešerši jsou využité knihy převážně novějšího data vydání, včetně několika novinek o Silverlight 5, ale jsou zde však i starší knihy, které svou kvalitou a obsahem vyvažují dřívější datum vydání.

Dalším tématem, které diplomová práce obsahuje je praktické využití technologií WPF a Silverlight v praxi. V případě technologie Silverlight jsou popsány aplikace provozované přímo společnostmi Microsoft, tak i komerční HeyGov! a produkty společnosti TescoSW a.s. U technologie WPF byl problém dohledat konkrétní příklady aplikací, které využívají tuto technologii. Jednou z nejznámějších komerčních aplikací je pravděpodobně Visual Studio 2010, mimo ni je zde popsána aplikace Family.Show, sloužící jako prezentace širokých možností WPF.

K analýze využitelnosti těchto technologií jsou využity data z Google Trends, StatOwl.com a Indeed.com. Na základě těchto dat je provedena analýza poptávky a využívání technologií WPF a Silverlight.

Testové otázky pro studenty jsou řešeny prostřednictvím testovacího softwaru, který obsahuje čtyřicet otázek zaměřených na technologie WPF a Silverlight. Po ukončení testu se zobrazí počet správných, nezodpovězených a špatně zodpovězených otázek. Následně si student může celý test projít a zkontrolovat u kterých otázek udělal chybu.

Posledním tématem diplomové práce je studijní opora přibližující čtenářům technologie WPF a Silverlight prostřednictvím aplikace, jejíž tvorba je v práci popsána.

Závěry této práce byly prezentovány na konferenci Internet, bezpečnost a konkurenceschopnost organizací 27. - 28. března 2012 konané na Univerzitě Tomáše Bati ve Zlíně.

ZÁVĚR V ANGLIČTINĚ

The thesis contains a literature search, containing information about WPF and Silverlight technologies. Resources are book format and in electronic form. In search are used books more recent releases, including some news about Silverlight 5, but there are used few older books, but its quality and content balancing earlier release date.

Next issue of this thesis is practical application of WPF and Silverlight technologies in practice. For Silverlight, there are described applications of Microsoft company and commercial HeyGov! and products of TescoSW a.s. For the WPF technology was difficult find specific examples of application. One of the most famous commercial application is Visual Studio, next application is described is Family.Show, it presents the possibility of the WPF.

To analyze the usability of these technologies are used data from Google Trends, StatOwl.com and Indeed.com. On this website information is available by searching terms on Google, information about the amount of jobs offered by the specified term or market research information in online trends.

Test questions for students are available in the testing software, which contains forty questions focused on WPF and Silverlight technologies, after the test ends, the number of correct, unanswered and wrong answers. Then the student can pass the test and review questions where he made a mistake.

The last topic of the thesis is to education support, approaching the possibility of WPF and Silverlight technologies to readers by the application, which a step by step.

The conclusions of this work were presented at the conference Internet, competitiveness and organizational security 27 - 28 March 2012, Tomas Bata University in Zlín.

SEZNAM POUŽITÉ LITERATURY

- [1] LAIR, Robert. Beginning Silverlight 4 in C#. 2010. United States of America: Apress, 2010. ISBN 978-1-4302-2988-9.
- [2] GHODA, Ashish. Introducing Silverlight 4. United States of America: Apress, 2010. ISBN 978-1-4302-2991-9.
- [3] MACDONALD, Matthew. Pro WPF in C# 2010: Windows Presentation Foundation in .NET. United States of America: Apress, 2010. ISBN 978-1-4302-7205-2.
- [4] SELLS, Chris a Ian GRIFFITHS. Programing WPF: Building Windows UI with Windows Presentatio Foundation. United States of America: O'Reilly Media, 2007. 2nd Edition. ISBN 0-596-51037-3.
- [5] ANDRADE, Chris, Shaw LIVERMORE, Mike MEYERS a Scott VAN VLIET. Professional WPF Programming: .NET Development with the Windows Presentation Foundation. United States of America: Wiley Publishing, 2007. ISBN 978-0-470-04180-2.
- [6] GUADIO SO, Victor. Foundation Expression Blend 4 with Silverlight. United States of America: Apress, 2010. ISBN 978-1-4302-2973-5.
- [7] MACDONALD, Matthew. Pro Silverlight 5 in C#. United States of America: Apress, 2012. ISBN 978-1-4302-3479-1.
- [8] MOSER, Christian. Logical- and Visual Tree. In: WPF Tutorial.net [online]. 2011 [cit. 2012-02-25]. Dostupné z: <http://wpftutorial.net/LogicalAndVisualTree.html>
- [9] TROELSEN, Andrew. Pro Expression Blend 4. United States of America: Apress, 2011. ISBN 978-1430233770.
- [10] MICROSOFT. MSDN [online]. 2012 [cit. 2012-03-17]. Dostupné z: <http://msdn.microsoft.com>
- [11] Stack Overflow [online]. 2012 [cit. 2012-03-17]. Dostupné z: <http://stackoverflow.com/>
- [12] The Code Project [online]. 2009 [cit. 2012-03-17]. Dostupné z: <http://www.codeproject.com>

- [13] SEDLÁK, Jan. Vancouver: Olympiáda Twitteru a Silverlightu. In: Živě [online]. 8. 2. 2010 [cit. 2012-04-14]. Dostupné z: <http://www.zive.cz/clanky/vancouver-olympiada-twitteru-a-silverlightu/sc-3-a-150798/default.aspx>
- [14] Média a technologie - Olympijské internetové vysílání ČT okusily tři miliony diváků. In: Česká televize [online]. 15. 3. 2010 [cit. 2012-04-14]. Dostupné z: <http://m.ceskatelevize.cz/ct24/media-a-technologie/83934-olympijske-internetove-vysilani-ct-okusily-tri-miliony-divaku/>
- [15] NBC Universal. In: Microsoft Case Study [online]. 5.11.2010 [cit. 2012-04-15]. Dostupné z: http://www.microsoft.com/casestudies/Case_Study_Detail.aspx?casestudyid=4000007258
- [16] TESCO SW a.s. Tesco SW a.s.: Výrobce a dodavatel specializovaných SW řešení a IS služeb [online]. 2012 [cit. 2012-04-15]. Dostupné z: <http://www.tescosw.cz/>
- [17] LARDINOIS, Frederic. Bing Maps Goes Silverlight. In: ReadWriteWeb [online]. 19.1.2010 [cit. 2012-04-15]. Dostupné z: http://www.readwriteweb.com/archives/bing_maps_silverlight.php
- [18] Bing Maps in HeyGov!: Helps Governments Manage 311 Issues. In: Bing [online]. 10.4.2010 [cit. 2012-04-15]. Dostupné z: http://www.bing.com/community/site_blogs/b/maps/archive/2010/10/04/bing-maps-in-heygov-helps-governments-manage-311-issues.aspx
- [19] MACDONALD, Matthew. Pro Silverlight 5 in VB. USA: Apress, 2012. 4th Edition. ISBN 978-1-4302-3518-7.
- [20] MACDONALD, Matthew. Pro WPF in VB 2010: Windows Presentation Foundation in .NET 4. USA: Apress, 2010. ISBN 978-1-4302-7240-3.
- [21] SUR, Abhishek. WPF Tutorial: A Beginning : 1. In: .NET Tutorials for Beginners and Intermediate level professionals [online]. 5.8.2010 [cit. 2012-04-19]. Dostupné z: <http://www.dotnetfunda.com/articles/article882-wpf-tutorial-a-beginning-1.aspx>
- [22] JavaFX: Rich Internet Applications Development [online]. 2011 [cit. 2012-04-19]. Dostupné z: <http://www.javaafx.com/>
- [23] Adobe [online]. 2012 [cit. 2012-04-19]. Dostupné z: <http://www.adobe.com/>

- [24] StatOwl.com: Statistical analysis and market research of Internet usage trends [online]. 2012 [cit. 2012-04-19]. Dostupné z: <http://www.statowl.com/>
- [25] Josh Smith on WPF [online]. 2012 [cit. 2012-04-19]. Dostupné z: <http://joshsmithonwpf.wordpress.com/>
- [26] Job Search: one search. all jobs. Indeed.com [online]. 2012 [cit. 2012-04-19]. Dostupné z: <http://www.indeed.com/>
- [27] Google Trends [online]. 2008 [cit. 2012-04-19]. Dostupné z: <http://www.google.com/trends/>
- [28] Visual Studio Magazine [online]. 2012 [cit. 2012-05-01]. Dostupné z: <http://visualstudiomagazine.com/>
- [29] Family.Show [online] [cit. 2011-05-01]. Dostupné z: <http://familyshow.codeplex.com/>

SEZNAM POUŽITÝCH SYMBOLŮ A ZKRATEK

XAML	Extensible Application Markup Language
WPF	Windows Presentation Foundation
WPF/E	Windows Presentation Foundation / Everywhere
VB	Visual Basic
RIA	Rich internet application
CLR	Common Language Runtime
DOM	Document Object Model
LOB	Line of bussines
IIS	Internet Information Services
SDK	Software development kit
UI	User interface
LGPL	Lesser General Public License
.NET	Prostředí pro běh aplikací a služeb

SEZNAM OBRÁZKŮ

OBR. 1. VISUAL TREE.....	14
OBR. 2. DESCKTOPOVÉ VS WEBOVÉ APLIKACE.....	16
OBR. 3. SILVERLIGHT APLIKACE.....	17
OBR. 4. BEGINNING SILVERLIGHT 4 IN C#	22
OBR. 5. INTRODUCING SILVERLIGHT.....	23
OBR. 6. PRO SILVERLIGHT 5 IN C#.....	24
OBR. 7. PRO SILVERLIGHT 5 IN VB.....	25
OBR. 8. PRO WPF IN C# 2010:WPF IN .NET 4	26
OBR. 9. PROGRAMING WPF	27
OBR. 10. PROFESSIONAL WPF PROGRAMMIN.....	28
OBR. 11. PRO WPF IN VB 2010	29
OBR. 12. FOUNDATION EXPRESSION BLEND 4.....	30
OBR. 13. PRO EXPRESSION BLEND 4	31
OBR. 14. MSDN.....	31
OBR. 15. XAML	32
OBR. 16. C# A VB	32
OBR. 17. STACK OVERFLOW	32
OBR. 18. THE CODE PROJECT.....	33
OBR. 19. SMOOTH STREAMING	35
OBR. 20. TESCOSW A.S. - FAMA+	37
OBR. 21. TESCOSW A.S. - MONIT	38
OBR. 22. TESCOSW A.S. - FAMA SERVICES	38
OBR. 23. BIND MAPS.....	39
OBR. 24. HEYGOV! - MIAMI	40
OBR. 25. HEYGOV! - SAN FRANCISCO.....	40
OBR. 26. JAVAFOX.....	43
OBR. 27. ADOBE FLASH.....	43
OBR. 28. STATISTIKA PODPORY PLATFORM U UŽIVATELŮ.....	44
OBR. 29. NAINSTALOVANÉ VERZE SILVERLIGHT	45
OBR. 30. NABÍDKY PRÁCE PRO INTERNETOVÉ TECHNOLOGIE.....	45
OBR. 31. GOOGLE TRENDS - SILVERLIGHT, FLASH A JAVAFOX.....	46
OBR. 32. GOOGLE TRENDS BEZ FLASHE	46
OBR. 33. JOB TRENDS - QT, WPF, WINFORMS	47
OBR. 34. GOOGLE TRENDS - QT, WPF, WINFORMS	48
OBR. 35. OTÁZKA V XML FORMÁTU PO SERIALIZACI.....	50
OBR. 36. STRUKTURA TŘÍDY	50
OBR. 37. APLIKACE PRO TESTOVÁNÍ ZNALOSTÍ	51

OBR. 38. TVORBA PROJEKTU VE VISUAL STUDIU.....	53
OBR. 39. STUKTURA NOVĚ VYTVOŘENÉHO PROJEKTU.....	54
OBR. 40. TVORBA GRAFICKÉHO ROZHRAŇÍ.....	54
OBR. 41. NASTAVOVÁNÍ VLASTNOSTÍ.....	55
OBR. 42. VÝSLEDNÁ APLIKACE HELLO WORLD.....	55
OBR. 43. ROZVRŽENÍ UI APLIKACE	59
OBR. 44. APLIKACE S VYPLNĚNÝMI ÚDAJI.....	63

SEZNAM TABULEK

TAB. 1. SLEDOVANOST ZOH 2010	35
TAB. 2. STATISTIKA PODPORY PLATFOREM U UŽIVATELŮ	44

