

**ČESKÁ ZEMĚDĚLSKÁ UNIVERZITA
PROVOZNĚ EKONOMICKÁ FAKULTA
Katedra informačních technologií**



DIPLOMOVÁ PRÁCE

**XML a OpenDocument Format standardy a jejich použití
v praxi**

**Autor práce:
Vedoucí diplomové práce:**

**Bc. Jan Dostál
Ing. Čestmír Halbich CSc.**

Česká zemědělská univerzita v Praze

Provozně ekonomická fakulta

Katedra informačních technologií

Akademický rok 2009/2010

ZADÁNÍ DIPLOMOVÉ PRÁCE

Jan Dostál

obor Informatika

Vedoucí katedry Vám ve smyslu Studijního a zkušebního řádu ČZU v Praze čl. 17 odst. 2 určuje tuto diplomovou práci.

Název práce: **XML a OpenDocument Format standardy a jejich použití v praxi**

Osnova diplomové práce:

1. Úvod
2. Cíl práce a metodika
3. XML
4. OpenDocument Format
5. Použití v praxi
6. Závěr
7. Seznam použitých zdrojů
8. Přílohy

Rozsah hlavní textové části: 60 - 80 stran

Doporučené zdroje:

GRUSOVÁ, Lucie. XML pro úplné začátečníky. ComputerPress, Brno 2002. 196s. ISBN: 80-7226-697-7.

KOSEK, Jiří. XML pro každého. Grada Publishing, Praha 2000. 164s. ISBN 80-7169-860-1.

LAURENT, Simon, FITZGERALD, Michael. XML Pocket Reference, Third Edition.

O'Reilly Media, 2009. 176s. ISBN 978-0-596-10050-6.

MLÝNKOVÁ, Irena, et al. Technologie XML - Principy a aplikace v praxi. Grada Publishing, Praha 2008. 272s. ISBN 978-80-247-2725-7.

SKONNARD, Aaron, GUDGIN Martin. XML pohotová referenční příručka. ComputerPress, Brno 2005. 344s. ISBN 80-247-0972-4.

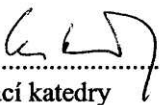
<http://www.w3.org/XML/>

<http://www.oxygenxml.com>

<http://www.opendocument.cz>

Vedoucí diplomové práce: **Ing. Čestmír Halbich, CSc.**

Termín odevzdání diplomové práce: duben 2011


.....
Vedoucí katedry




.....
Děkan

V Praze dne: 15. 1. 2010

Čestné prohlášení:

Já Jan Dostál prohlašuji, že jsem diplomovou práci na téma „XML a OpenDocument Format standardy a jejich použití v praxi“ po odborných konzultacích s vedoucím práce Ing. Čestmírem Halbichem CSc. vypracoval samostatně za použití uvedené literatury. Tímto prohlášením dávám rovněž souhlas k eventuelnímu zveřejnění mé diplomové práce v elektronické či tištěné podobě.

V Praze dne 5.4. 2011

.....

(podpis autora práce)

Poděkování:

Tímto chci velmi poděkovat především vedoucímu diplomové práce, panu inženýru Čestmíru Halbichovi CSc., za podporu a velmi dobrou spolupráci při tvorbě diplomové práce. Dále bych chtěl poděkovat panu Bc. Danielovi Petržílkovi, zaměstnanci firmy Software602 a.s., který mi pro účely tvorby této diplomové práce poskytl testovací certifikát k programu 602XML Designer.

Jan Dostál

XML a OpenDocument Format standardy a jejich použití v praxi

Souhrn

Diplomová práce se zabývá standardem XML. V první části je popsána jeho historie spolu s klíčovými vlastnostmi. Navazuje druhá část popisující schopnosti přidružených technologií. Třetí část se zabývá formátem ODF, který je na XML založen.

Následuje praktická část s komplexními příklady, které využívají schopností XML. V této části lze nalézt také porovnání formátu ODF s konkurencí. V závěru jsou poté poznatky vyplývající z praktické části popsány.

Klíčová slova

XML, OpenDocument Format, ODF, DTD, XPath, XSLT, XPointer, XLink, 602Designer, jmenné prostory

Summary

The thesis deals with the XML standard. The history and the key features of the XML standard are described at the first part of the thesis. At the second part we can find the description of the associated abilities. The third part is about the ODF format, which is based on the XML.

The practical part contains the complete examples which use the XML abilities. In this part we also can find the comparison of the ODF format and the competition. The conclusion describes the knowledge which came from the practical part.

Key words

XML, OpenDocument Format, ODF, DTD, XPath, XSLT, XPointer, XLink, 602Designer, namespaces

Obsah:

1. Úvod.....	9
2. Cíle a metodika	10
2.1 Cíl práce	10
2.2 Metodika	10
3. XML.....	11
3.1 Historie.....	11
3.2 Vlastnosti XML	12
3.3 Syntaxe XML.....	14
3.4 DTD	15
3.4.1 Deklarace elementů.....	16
3.4.2 Deklarace atributů.....	19
3.5 Jmenné prostory.....	22
3.6 XPath	23
3.7 XLink.....	27
3.7.1 Jednoduché odkazy	27
3.7.2 Rozšířené odkazy	28
3.7.3 XPointer	28
3.8 Stylovací jazyky.....	30
3.8.1 Připojení stylu k dokumentu	31
3.8.2 XSL.....	32
3.8.2.1 XSLT	32
3.8.2.1.1 Řazení	33
3.8.2.2 XSL:FO.....	34
3.8.2.2.1 Struktura XSL:FO dokumentu.....	35
4. OpenDocument Format.....	36
4.1 OASIS	36
4.2 Historie.....	37
4.3 Proč používat ODF	39
4.3.1 OpenDocument je standard.....	39
4.3.2 Otevřený standard	40
4.3.3 Podpora	41
4.3.4 Státní správa.....	42
4.3.5 Firemní prostředí.....	43
4.3.6 Školství, věda a výzkum	45
4.3.7 Domácnosti a malé firmy.....	46
5. Použití v praxi	47
5.1 XML.....	47
5.1.1 DTD	47
5.1.2 Zkušební XML.....	49
5.1.3 XSLT	51
5.1.3.1 Html výstupy.....	51
5.1.3.2 Výstup SQL	55
5.1.3.3 Výstup do formátu .fo	55
5.1.4 Zhodnocení XML	61

5.2 602XML Designer	62
5.2.1 Tvorba formuláře	62
5.2.2 Výsledné formuláře.....	68
5.2.2 Zhodnocení 602XML Designer	70
5.3 Porovnání kancelářských formátů	71
5.3.1 Výběr konkurence.....	71
5.3.2 Porovnání editorů a formátů	72
5.3.3 Dotazník.....	75
5.3.4 Zhodnocení ODF	75
6. Závěr	77
7. Seznam použitých zdrojů.....	79
7.1 Seznam obrázků.....	81
7.2 Seznam tabulek	81
8. Přílohy.....	82

1. Úvod

Informace! Jedna z věcí, bez kterých by dnešní společnost nemohla fungovat. Je jich téměř nekonečné množství, přesto mají cenu zlata.

Již od počátků civilizace potřebují lidé ke svému životu určité vědomosti a poznatky - informace. Jak se rozvíjí věda a technika, vyplouvají na povrch informace stále nové a nové, s čímž souvisí potřeba je nějakým způsobem zaznamenávat a zpracovávat.

Ještě v nedávné minulosti byly nejdůležitějším zdrojem informací psané, později tištěné, dokumenty. V minulém století se však objevilo nové informační médium- Internet. V době svého vzniku byl Internet dostupný pouze akademickým a vládním organizacím, avšak technologický rozvoj umožnil jeho rozšíření mezi běžné uživatele. Internet nabídl do té doby nevídané a neomezené možnosti v uchovávání a sdílení informací. Žádná světová knihovna se v počtu dostupných informací nemůže s internetem rovnat.

Aby bylo možné s takovým množstvím informací pracovat, je potřeba mít k dispozici efektivní způsoby sdílení a vyhledávání. Před příchodem XML se k tomu používaly jen proprietární technologie, které pracovaly pouze s úzkým okruhem aplikací. Výměna dat mezi informačními systémy proto byla velmi nákladná a neefektivní.

Jazyk XML je otevřená technologie, která do této oblasti přinesla pokrok. U dokumentů popisuje strukturu z hlediska jejich významu, nezabývá se vzhledem.

Hlavním tématem této diplomové práce je vysvětlit výhody XML, popsat jeho funkce a vše ukázat na praktickém příkladu.

2. Cíle a metodika

2.1 Cíl práce

Hlavním cílem této práce je popsat problematiku sdílení dokumentů, informovat o standardech XML a ODF, jejich základních vlastnostech a možnostech jejich využití.

Dále je cílem práce určit silné a slabé stránky XML, u případných slabých stránek poté nalézt možnosti jejich eliminace.

Posledním cílem práce je porovnat Opendocument Format s konkurenčními formáty a doporučit některý formát k používání.

2.2 Metodika

Použitá metodika diplomové práce:

- Zadání tématu diplomové práce
- Vytyčení cílu diplomové práce
- Vypracování osnovy
- Zajištění odborné literatury týkající se zvoleného tématu
- Vypracování literární rešerše
- Vytvoření praktického projektu
- Analýza tvorby projektu
- Dotazníkový průzkum
- Porovnání ODF s konkurencí
- Analýza porovnávání
- Celkové zhodnocení
- Konzultace s vedoucím diplomové práce

3. XML

Extensible Markup Language, čili XML, je obecný značkovací jazyk vyvinutý konsorciem W3C.

3.1 Historie

Prvním známým značkovacím jazykem byl GML, vytvořený při práci na systému pro uchování a následné využívání právních textů IBM. Autoři si tehdy museli poradit s nekompatibilitou jednotlivých systémů, nejsnazší cesta vedla právě přes vytvoření obecného značkovacího jazyka.

Princip jazyka GML se ukázal funkčním a tak v osmdesátých letech začala na jeho základě vyvíjet standardizační organizace ANSI jazyk, umožňující definici vlastních značkovacích jazyků. Uživatel si mohl vytvořit vlastní sadu značek hodících se pro právě používaný dokument. Již dříve vytvořilo sdružení GCA formátovací jazyk použitelný na širokém spektru zařízení- GenCode. Mnohé cíle obou projektů byly podobné a tak došlo k jejich spojení. Výsledkem se stal jazyk SGML (Standard Generalized Markup Language), definovaný v ISO normě 8879 z roku 1986.

Jazyk SGML je až moc obecný. Splnil cíl, umožňuje definici vlastních značkovacích jazyků čili sad značek a vztahů mezi jimi, pomocí definice typu dokumentu, DTD. K tomu má i spoustu volitelných parametrů, jako například maximální délka názvu značek, určení znaku použitelného jako oddělovače značek od textu. Tato veliká obecnost standardu SGML zbrzdila jeho praktické využití.

Velkou podporu pro jazyk SGML znamenalo americké ministerstvo obrany, které od svých dodavatelů vyžadovalo dokumentaci právě ve formátu SGML. Důvod byl zřejmý- bylo třeba, aby dokumentace byla použitelná v poměrně dlouhém období. Nebylo tedy možné použít nějaký proprietární formát textového procesoru, který se každých pár let mění. [3]

Určitě nejznámější aplikací jazyka SGML je pak jazyk HTML používaný pro tvorbu webových stránek. Značky použitelné na stránkách určuje DTD, které se pro různé verze HTML mění.

Díky své jednoduchosti si jazyk HTML získal v polovině devadesátých let velkou podporu. Později se však ukázalo, že skupina značek, které HTML používá už nestačí. Pro

účely vyhledávání a efektivnější výměny dat byly potřeba vlastní značky, které by přesněji vymezily význam textu. To by hravě zvládnul komplexní jazyk SGML

Jak již bylo řečeno, standart SGML je velmi komplexní a jeho úplná implementace náročná. Při praktickém používání se však ukázalo, že se používá jen část jeho možností. Z SGML se tedy vybrala tato malá část, XML.

Narozdíl od SGML je v XML mnoho parametrů, jako například oddělovače, předem nastaveno, zůstala však možnost definování vlastních DTD, tedy i vlastních elementů. Syntaxe dokumentů XML je oproti SGML velmi přísná, díky čemuž je vývoj aplikací schopných s tímto formátem pracovat snazší a levnější.

3.2 Vlastnosti XML

XML je otevřený formát, na webu konsorcia W3C je jeho přesná specifikace volně k dispozici. Každý tak může podporu XML implementovat do svých aplikací zcela zdarma, bez toho aby byl svazován nějakými licenčními podmínkami[1].

Otevřenost formátu znamená mimo jiné, že není svázán s konkrétní aplikací, firmou, dokonce ani platformou.

Narozdíl od mnoha ostatních není XML formát binární, je založený na obyčejném textu. Jeho tvorbu, či editaci tak lze v případě potřeby provést v libovolném textovém editoru.

Formát XML již od počátku vývoje dbá na potřeby jiných jazyků, než je angličtina. Jako znaková sada se používá ISO 10646, což je 32bitová znaková sada, která dokáže pojmout všechny znaky dnes používaných jazyků.

Psaní v ISO 10646 by však například při použití češtiny bylo zbytečným plýtváním místem. Dokumenty mohou být psány v libovolném kódování, které je v každém dokumentu přesně určeno.

Při používání XML je potřeba uložené informace zobrazit. Například na webu, či vytisknout na papír. V takových případech je třeba přesně definovat, jak se obsah jednotlivých elementů zobrazí. K tomu však XML žádné prostředky neobsahuje. Tyto prostředky je nutné nalézt v některém ze stylových jazyků. Soubor pravidel a příkazů,

kteřé říkají jak se dokument transformuje do jiného, například zobrazitelného, formátu se nazývá styl.

Vytvořený styl lze aplikovat na více dokumentů stejného typu, což zajistí jejich jednotné formátování. Zároveň je možné na jeden dokument aplikovat několik různých stylů, vytvořit tak například webovou stránku, soubor pro tisk i příkaz pro vložení dat do databáze.

Nejznámější ze stylových jazyků je patrně jazyk CSS, který se používá i u formátování webových stránek. CSS však lze využít jen pro jednoduché formátování, sloužící k zobrazení na obrazovce. Pro náročnější operace slouží jazyk XSLT, který dokáže dokument před samotným formátováním transformovat.

Dokumenty vytvářené pomocí jazyka XML nabízejí dvě možnosti automatické kontroly. První z nich je takzvaná kontrola správného zformátování. Tato kontrola kontroluje obecné zásady pro psaní XML, které jsou stejné pro všechny XML dokumenty. Tyto zásady jsou:

- Obsah elementu musí začínat počátečním tagem. <element>
- Konec elementu je označen koncovým tagem. </element>
- Před ukončením elementu musí být ukončeny všechny jeho vnitřní elementy.
- Počáteční tag může obsahovat atributy. <element jmeno="element1">
- Prázdný element se zapisuje <element />.
- Pravidla pro psaní speciálních znaků, například "<".

Druhý typ kontroly se nazývá kontrola požadavků platnosti. Tato kontrola ověřuje, zda vytvořený dokument splňuje podmínky stanovené v připojeném DTD. Programu, který se stará o kontrolu správnosti XML dokumentů se říká parser. Funkční parser je dnes součástí například webových prohlížečů.

XML umožňuje vytváření odkazů v rámci jednoho dokumentu i mezi dokumenty navzájem. Nabízí však mnoho možností nad rámec odkazů, které jsou známé z HTML. Lze vytvářet i vícesměnné odkazy, které spojují několik dokumentů dohromady. Užitečná je i možnost uložení odkazů zcela mimo dokumenty, kterých se týkají. Tímto způsobem lze vytvářet různé anotace a komentáře k již existujícím stránkám. [3]

Tvorba odkazů je dnes popsána ve třech standardech XLink, Pointer a XPath. XPath je jazyk, který umožňuje adresovat jednotlivé části dokumentu. Jeho možnosti dále rozšiřuje jazyk XPointer. XPointer se používá k určování jednotlivých částí dokumentu ve stylu: zajímá mě první odstavec třetí kapitoly. Není proto třeba všechny části dokumentu, na které chceme odkazovat, explicitně označovat pomocí návěští jako v HTML. [3]

XLink je jazyk pro tvorbu odkazů. Jednotlivé dokumenty se určují pomocí jejich URL adresy, za kterou lze uvést ještě Pointer pro přesnější určení části dokumentu. [3]

3.3 Syntaxe XML

Každý XML dokument se skládá z elementů, které jsou do sebe vzájemně vnořeny. V textu se elementy označují pomocí tagů. Většina elementů je označena pomocí počátečního a koncového tagu. Názvy elementů se zapisují mezi znaky < >, koncový tag obsahuje navíc znak /.

<element>Obsah elementu</element>

Lze použít i elementy, které nemají žádný obsah (například obrázek). Prázdný element lze zapsat *<obrazek/>*.

XML dokument musí pro každý počáteční tag obsahovat i odpovídající tag ukončující. Před ukončením elementu musí být ukončeny i všechny jeho vnitřní elementy. Čili každý element musí mít počáteční i koncový tag uvnitř stejného elementu. Z toho tedy plyne, že každý XML dokument musí být celý obsažený v jednom elementu.

Elementy jsou základním stavebním kamenem XML dokumentů, u každého počátečního tagu navíc mohou (v závislosti na DTD někdy musí) být uvedeny atributy elementu.

<element atribut="zkusebni"></element>

Jelikož se znaky "<" a ">" používají pro oddělení tagů, nelze je do dokumentu zapsat. Pro zápis těchto znaků se tedy používají takzvané znakové entity. Znak "<" lze zapsat <, znak ">" poté >. Pro zápis ampersandu se poté používá entita &.

Na další ukázce je vidět, jak do dokumentu vložit komentář.

<!--Komentář -->

Komentář je součástí dokumentu, avšak parser jej ignoruje, není zpracováván. Komentář může obsahovat všechny znaky, kromě sekvence znaků „--“. Může obsahovat i tagy, což se hodí při editaci dokumentu.

Takto naformátovaný dokument splňuje požadavky XML, je však vhodné aby obsahoval ještě hlavičku s verzí XML a druhem použitého kódování.

```
<?xml version="1.0" encoding="windows-1250"?>
```

Všechny aplikace podporující XML by měly bezchybně pracovat se standardním kódováním UTF-8.

3.4 DTD

V přehledu vlastností XML jsem se již zmínil o kontrole platnosti dokumentu, čili o splnění podmínek zadaných v DTD (Document Type Definitinon).

V DTD je nejen definováno jaké elementy a příslušné atributy lze v dokumentu použít, ale i vzájemné vztahy mezi elementy. DTD je tedy užitečný nástroj hlídající, že dokumenty, s kterými pracujeme, mají požadovanou strukturu. Lze použít ji vytvořené DTD jako například HTML, či rozšířený DocBook, který definuje elementy vhodné pro značkování technické literatury. Existují různé skupiny a sdružení, která vydávají DTD pro použití v určitých oblastech. Namátkou například server XML.org provozovaný sdružením OASIS. Při použití vytvořeného DTD nejde o nic jiného, než se shodnout na názvech několika elementů, které se budou standardně používat pro označování určitých částí dokumentu. V současnosti existují rozmanité sady značek.

Výhodou XML je použití jmenných prostorů (namespaces), díky kterým může být jedna část dokumentu založena na standardním DTD, pro druhou pak lze použít DTD vlastní[1].

Při použití nějakého standardního DTD získáme možnost využívat již vytvořené jednoúčelové nástroje, které nám usnadní práci.

Pokud chceme v dokumentu využít výhod DTD, je potřeba použité DTD určit deklarací typu dokumentu, která se umísťuje na začátek, zpravidla na druhá řádek pod deklarací verze XML a použitého kódování. Aby bylo DTD možné použít na více dokumentů, bývá umístěno ve zvláštním souboru. Deklarace vypadá:

```
<!DOCTYPE prace SYSTEM "dtd/prace.dtd">
```

Ve vzorové deklaraci „prace“ uvádí kořenový element dokumentu, „dtd/práce.dtd“ pak cestu k DTD souboru.

Při použití veřejného identifikátoru pro některé standardní DTD se slovo „SYSTEM“ nahradí slovem „PUBLIC“. Do uvozovek poté vložíme veřejnou adresu DTD. I v tomto případě však připojíme i lokální cestu pro případ, že by veřejné DTD nebylo přístupné.

```
<!DOCTYPE prace PUBLIC “ // Norman Walsh/DTD docbook XML V4”  
“docbook.dtd”>
```

Při publikování dokumentů mimo lokální síť by cesta “docbook.dtd” samozřejmě musela být nahrazena příslušnou URL.

Další možností, jak použít DTD je vložit jej přímo do dokumentu.

```
<!DOCTYPE prace [  
zde se bude nacházet DTD  
]>
```

Takovéto použití není příliš časté, protože díky tomu přicházíme o možnost sdílení DTD mezi různé dokumenty. Avšak v některých případech se používá kombinace obou předešlých řešení. Jelikož lokální část DTD se zpracovává před tou externí, lze tak použít externí DTD a upravit ho pomocí lokálně uvedených definic.

```
<!DOCTYPE prace SYSTEM “dtd/prace.dtd”[  
zde se bude nacházet DTD lokálně upravující prace.dtd  
]>
```

3.4.1 Deklarace elementů

Deklarace nového elementu je velice snadná.

```
<!ELEMENT nazev_elementu (obsah_elementu)
```

Název elementu musí začínat písmenem, další znaky názvu pak mohou obsahovat písmena, číslice i některé speciální znaky jako “.”, “-”, “_”, “:”. Omezení neplatí jen na písmena anglické abecedy, jak bývá v počítačových systémech zvykem. V názvech lze použít diakritika, azbuka nebo třba hebrejšťina. [3]

Délka názvu není omezena, narozdíl od HTML a SGML však je důležitá velikost písmen.

Následující příklad ukazuje deklaraci dvou různých elementů- “nadpis” a “Nadpis”.

```
<!ELEMENT nadpis obsah_elementu>
```

```
<!ELEMENT Nadpis obsah_elementu>
```

Nejzajímavější je však část deklarace elementu “obsah_elementu”. Tato část definuje co může, případně musí, element obsahovat. Nejjednodušší je prázdný element, který nemůže obsahovat žádné další elementy, ani text. Takovým elementem může být například obrázek, či v HTML používaná značka
.

```
<!ELEMENT img EMPTY>
```

```
<!ELEMENT br EMPTY>
```

Anglické slovo “EMPTY” určuje, že element nesmí nic obsahovat. V dokumentu se takový element zapíše , nebo zkráceně . Nelze však zapsat pouze
 jako v HTML.

Mohlo by se zdát, že prázdný element “img” nelze k ničemu použít. V takovémto zápisu by opravdu neměl smysl, lze k němu však přiřadit například atribut url (viz další kapitola). Poté již půjde s obrázkem dále pracovat.

Opakem klíčového slova “EMPTY” je slovo “ANY”. Při použití “ANY” nijak neomezujeme obsah elementu. Ten tak může obsahovat libovolné další elementy i text. V praxi se “ANY” používá zejména při zkoušení rozpracovaných DTD.

Požadavky na jednotlivé elementy jsou však povětšinou mnohem složitější, než umožňují definovat klíčová slova ANY a EMPTY. V takových případech použijeme takzvanou modelovou skupinu. Modelová skupina se používá pro definici elementů, které obsahují další elementy nebo mají smýšlená obsah, čili obsahují již přímo text a elementy.

Modelová skupina je uzavřena do kulatých závorek a obsahuje minimálně jedno slovo. Tímto slovem bývá nejčastěji jméno elementu, který může být v právě definovaném elementu obsažen. Vnořené elementy je možné navzájem kombinovat pomocí oddělovačů AND “,” a OR “|”. Elementy oddělené čárkou tedy musí následovat v pořadí, v kterém

jsou uvedeny. Pokud má tedy kapitola obsahovat “nadpis” a poté “odstavec”, syntaxe bude vypadat následovně:

```
<!ELEMENT kapitola (nadpis, odstavec)>
```

Pokud jsou elementy odděleny znakem “|”, může být v dokumentu uveden pouze jeden z nich. Element “rodic” má obsahovat element “otec” nebo “matka”. Syntaxe následující:

```
<!ELEMENT rodic (otec | matka)>
```

Pomocí závorek lze oba operátory vzájemně kombinovat a tak například případ, kdy “kapitola” má obsahovat “nadpis1” nebo “nadpis2” a poté “odstavec” zapíšeme následujícím způsobem:

```
<!ELEMENT kapitola ((nadpis1 | nadpis2), odstavec)>
```

Pokud je v modelové skupině uveden pouze název elementu, musí být obsažen právě jednou. Kromě pořadí elementů lze určovat i jejich počet. Zda jsou povinné, či zda se mohou opakovat. Počet výskytů daného elementu lze určit následujícími symboly.

- symbol “+” znamená jeden, nebo více výskytů
- symbol “*” libovolné (i nulové) množství elementů daného názvu
- symbol “?” znamená žádný nebo jeden výskyt
- jak již bylo psáno- pouhý název elementu znamená právě jeden výskyt

Vše lze podle potřeby kombinovat. Indikátor počtu výskytů lze připojit i modelovou skupinu a tak při použití závorek získáváme možnost velmi přesné definice obsahu elementu.

Jestliže je obsahem elementu text, vyjádříme to slovem #PCDATA. Pokud by element „odstavec“ obsahoval pouze text a ne další elementy, syntaxe by vypadala:

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

Pokud má element smíšený obsah (další elementy a text), musí mít deklarace speciální tvar. #PCDATA musí být ve skupině uvedeno jako první, skupina musí být spojena pomocí operátoru „|“ a musí být volitelně opakovatelná. Zápis tedy vypadá:

```
<!ELEMENT odstavec (#PCDATA | img)*>
```

3.4.2 Deklarace atributů

V XML může mít kterýkoliv element libovolný počet atributů. Pomocí atributů se většinou k elementům připojují různé metainformace. Deklarace atributů má velice jednoduchý tvar:

```
<!ELEMENT název_elementu deklarace_atributů >
```

“*Název_elementu*” označuje element, pro který atributy deklaruje. “*Deklarace_atributů*” se potom skládá ze tří částí. První část je název atributu. Pro názvy atributů platí stejná omezení, jako pro názvy elementů, včetně rozlišování velkých a malých písmen. Druhou částí je typ atributu. Třetí, poslední, část určuje standardní hodnotu atributu, případně zda je jeho použití povinné.

Pokud je atribut povinný, uvede se za deklaraci jeho typu #REQUIRED. Atribut elementu je poté nutné zadat, v opačném případě ohlásí parser chybu.

Protipólem je poté situace, kdy atribut není potřeba zadat. V těchto případech se použije klíčové slovo #IMPLIED. Deklarace se opakuje pro každý atribut, který lze u elementu použít.

Základním typem atributu je CDATA, umožňující zadat jako hodnotu jakýkoliv textový řetězec. Deklarace tedy vypadá následovně:

```
<!ATTLIST autor username CDATA #REQUIRED>
```

V dokumentu se poté atribut zadá následovně:

```
<autor username=“xdosj301”>Jan Dostál</autor>
```

Dalším typem atributu je například NMTOKEN. Atribut tohoto typu může obsahovat jedno slovo, které se skládá z písmen, číslic a několika dalších znaků (stejně omezení jako u názvů elementů, či atributů). Podobným typem atributu je MNTOKENS. Atribut tohoto typu může obsahovat několik slov oddělených mezerou.

Deklarace v DTD tedy vypadá:

```
<!ATTLIST prace nazevsouboru NMTOKEN #REQUIRED>
```

```
<!ATTLIST prace klicova_slova MNTOKENS #IMPLIED>
```

Pokud má element více atributů, lze to zapsat zkráceně takto:

```
<!ATTLIST prace    nazevsouboru NMTOKEN #REQUIRED
                  klicova_slova NMTOKENS #IMPLIED>
```

Oba zápisy jsou ekvivalentní.

Zadání do XML dokumentu pak vypadá následovně:

```
<prace nazevsouboru="prace.xml" klicova_slova="XML DTD Xpath XSLT">
</prace>
```

Dalšími typy atributů jsou ID, IDREF a IDREFS, které se používají pro vytváření odkazů v rámci jednoho dokumentu. Pokud atribut definujeme jako ID, musí mít v rámci dokumentu přiřazenou jedinečnou hodnotu. Přitom všechny atributy typu ID sdílí stejný prostor. Omezení na jedinečnost platí i pro atributy s různým názvem u různých elementů. Pokud je pravidlo jedinečnosti porušeno, ohlásí parser chybu.[3]

Atributy s typem IDREF pak mohou obsahovat pouze hodnotu použitou v některém atributu typu ID. Typ IDREFS umožňuje použít v jednom atributu více hodnot najednou. Podobně jako u NMOKENS se hodnoty oddělují mezerami.[3]

Mezi typy atributů patří rovněž NOTATION, sloužící k určení typu dat elementu, pokud data nejsou ve formátu XML.

Další možností, jak vymezit typ atributu je uvedení výčtu jeho hodnot. Jak vidět na následujícím příkladu, je to velmi snadné.

```
<!ATTLIST pisemka  hodnoceni (0|1|2|3|4|5) #REQUIRED>
```

Hodnocení písemky tedy může nabývat hodnot 0 / 1 / 2 / 3 / 4 nebo 5.

Místo vlastností #REQUIRED a #IMPLIED lze zadat standardní hodnotu. Tato hodnota se následně pro atribut použije pokud uživatel nezadá jinou.

Například:

```
<!ATTLIST pisemka  hodnoceni (0|1|2|3|4|5) "0">
```

Před standardní hodnotu lze ještě uvést #FIXED. Tím je možné určit, že atribut nebude mít nikdy jinou hodnotu, než standardní.

```
<!ATTLIST obrazek zarovnani CDATA "#FIXED=left">
```

Další typy atributů, které lze použít jsou ENTITY a ENTITYES, používané pokud atribut obsahuje jméno (ENTITY), nebo jména (ENTITYES) entit.

Parametrické entity používáme pokud v textu často pakujeme stejné sekvence textu. Pokud například více elementů má několik společných atributů. V tom případě je zbytečné opakovaně opisovat deklaraci stejných atributů. Práci usnadní parametrické entity. Použití parametrické entity z funkčního hlediska odpovídá napsání celého textu, šetří však práci.

Příklad použití parametrické entity:

```
<!ENTITY % cislice "(0|1|2|3|4|5|6|7|8|9) #REQUIRED">
<attlist hodnoceni    vzhled %cislice;
                      obsah %cislice;>
```

Element “hodnoceni” tak má povinně přiřazeny atributy “vzhled” a “obsah”. Jako hodnoty mohou být číslice 0 až 9.

druhý příklad:

```
<!ENTITY % spolecne_atributy    "email CDATA #IMPLIED
                                user_name CDATA #REQUIRED">
<attlist autor    %spolecne_atributy;>
<attlist vedouci    %spolecne_atributy;
                  kancelar CDATA #IMPLIED>
```

Společné atributy autora a vedoucího jsou “email” a “user_name”. Pro element “vedouci” je přidán ještě volitelný atribut “kancelar”.

Samozřejmě jsem nevyčerpal všechny možnosti, jak lze definovat strukturu dokumentu pomocí DTD, avšak snažil jsem se popsat ty důležité, často používané.

DTD slouží hlavně pro definování struktury textových dokumentů, například u databází je potřeba využít mocnější nástroje, které poskytují například XML Schémata.

3.5 Jmenné prostory

Často je velmi užitečné v jednom dokumentu používat několik sad elementů a jejich atributů, definovaných v různých DTD. Kromě vlastní sady značek tak mohou například za účelem snadnějšího vyhledávání použít ještě sadu již vytvořených a používaných značek. Takovou možnost nabízejí právě jmenné prostory (namespaces).

Pokud chceme u nějakého elementu, či elementů v něm vnořených, používat elementy z určité sady značek, je potřeba takovou možnost nastavit pomocí speciálního atributu „xmlns“.

```
xmlns:prefix_prostoru = "URI_adresa"
```

Jako „prefix_prostoru“ použijeme název, který chceme pro prostor používat, adresa poté značí cestu k DTD. Příklad použití:

```
<moje:text xmlns:moje="dtd/kniha.dtd"
           xmlns:book="http://www.xxx.com/book">
  <moje:autor username="xdosj301">Jan Dostal
</moje:autor>
  <book:isbn>54545</book:isbn>
  <book:title>XML</book:title>
</moje:text>
```

Pokud je jedna sada značek používaná častěji, lze jeden jmenný prostor nastavit implicitně, stačí vynechat jeho prefix.

```
<moje:text xmlns:moje="dtd/kniha.dtd"
           book="http://www.xxx.com/book">
  <moje:autor username="xdosj301">Jan Dostal
</moje:autor>
  <isbn>54545</isbn>
  <title>XML</title>
</moje:text>
```

Zápis je úspornější, což by bylo vidět hlavně u delší ukázky. U implicitního jmenného prostoru však nelze u elementů používat atributy. Proto by například prostor „moje“ nešel nastavit jako implicitní z důvodu použití atributu „username“ u „autora“.

3.6 XPath

XPath (XML Path Language) je samostatný standard W3C, sloužící k vybrání části XML dokumentu. XML dokument je přitom chápán jako stromová struktura, kde jsou jednotlivé elementy, atributy a text chápány jako uzly. Výsledkem jazyka XPath je hodnota jednoho z následujících typů:

- množina uzlů ze stromu vstupního dokumentu
- logická hodnota
- číslo
- textový řetězec

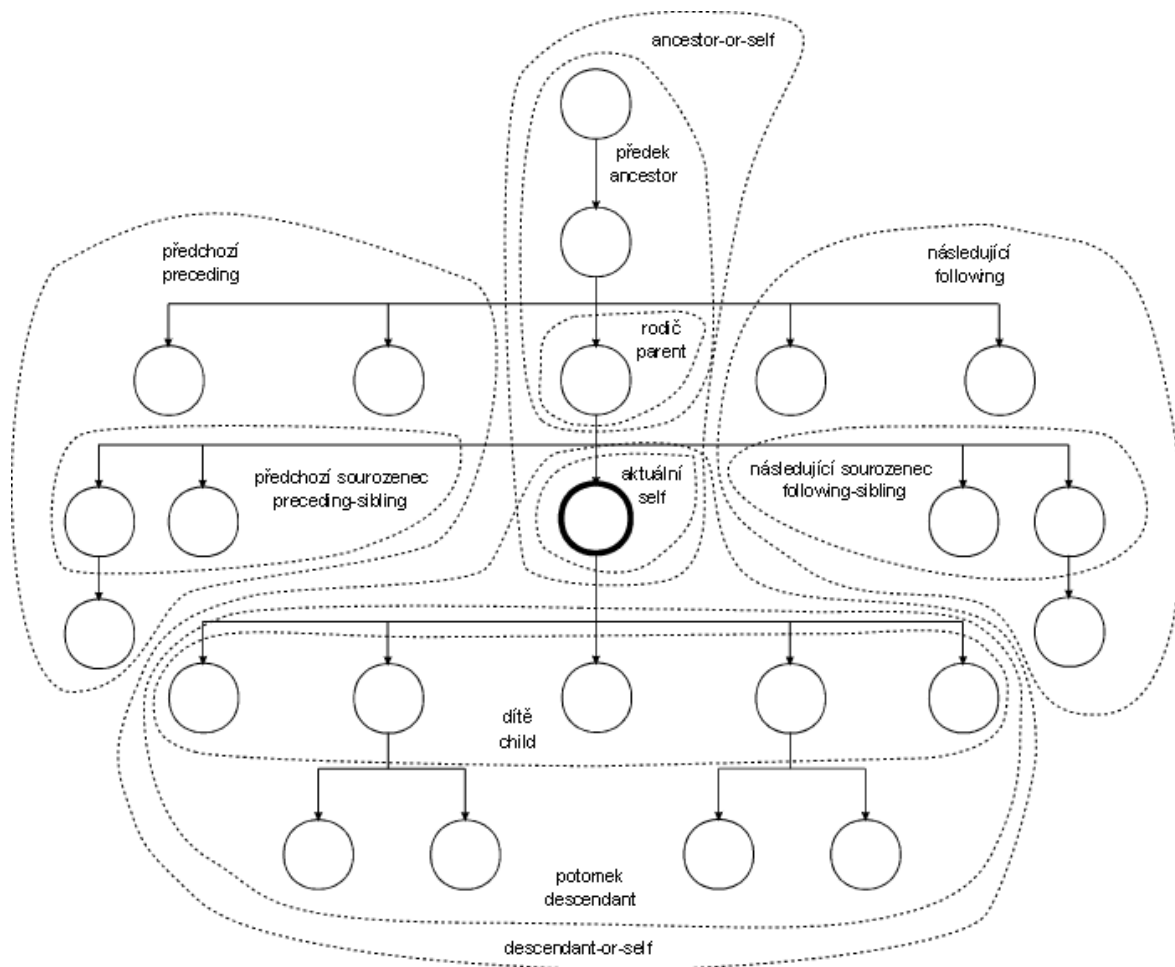
Základní součástí jazyka je výraz popisující cestu. Taková cesta se zapisuje jako posloupnost přechodů mezi jednotlivými sadami uzlů, oddělených lomítky. Každý přechod je určen pomocí tří složek (některé ovšem nemusí být uvedeny, pokud mají implicitní hodnotu): [18]

- osa
- test uzlu
- predikát

Specifikace osy určuje, ve kterém směru od aktuálního uzlu se bude vyhodnocení po stromové struktuře XML dokumentu pohybovat. Test uzlu umožňuje na základě názvu a typu vybírat jen některé uzly. Pomocí predikátů jsou na základě zadaných podmínek vybrané uzly dále testovány.

XPath definuje celkem třináct os. Sou jimi ancestor, ancestor-or-self, attribute, child, descendant, descendant-or-self, followin, following-sibling, namespace, parent, precedind, precedingl-sibling, self.

Významy jednotlivých os jsou vidět na obrázku č. 1.



Obrázek č. 1- Osy

Implicitní osou je osa child, pokud chceme použít jinou osu, je třeba jí oddělit dvojitou dvojtečkou. Testy uzlů na různých osách se však používají velmi často, proto zkrácené zápisy existují i pro jiné osy.

Například místo osy atributu lze použít zavináč.

`@id` je ekvivalentní zápis k `atribute::id`

Test uzlu umožňuje na vybrané ose vybrat jen některé uzly. Testovat lze název uzlu a jeho typ. Pokud chceme vybírat jiné uzly, než jsou elementy, atributy a jmenné prostory, lze použít ještě následující testy:

- `node()` – vybere všechny uzly
- `text()` – vybere textové uzly
- `processing-instruction()` – vybere všechny uzly odpovídající instrukci pro zpracování

V každé části cesty lze ještě použít několik predikátů, které zvýší podmínky pro vybrání uzlů. Predikáty se vyhodnocují pro každý prozatím vyhovující uzel, zapisují se do hranatých závorek.

XPath výrazy lze dále skládat pomocí logických spojek, či relačních a matematických operátů.

Pro úplnost ještě dodám, že v XPath lze používat i několik funkcí. Funkce se hodí v predikátech a v případech, kdy má XPath výraz vracet něco jiného než množinu uzlů.

Ukázky zápisu s popisem výběru jsou vidět v tabulce č. 1.

Výraz	Co vybere
*	všechny elementy, které jsou dětmi aktuálního uzlu
text()	všechny textové uzly, které jsou dětmi aktuálního uzlu
kapitola	všechny elementy „kapitola“, které jsou dětmi aktuálního uzlu
*/kapitola	všechny elementy „kapitola“, které jsou vnoučaty aktuálního uzlu
kapitola/*	Všechny děti elementu „kapitola“, který je dítětem aktuálního uzlu
id(kap1)	uzel, který má u atributu „id“ hodnotu „kap1“
kapitola[1]	první element „kapitola“, který je dítětem aktuálního uzlu
/dokument/kapitola[last()]	poslední element „kapitola“, díky lomítku na začátku není výraz vztažen k aktuálnímu, ale ke kořenovému uzlu, který má jediné dítě- kořenový element dokumentu
kapitola[3]/odstavec[4]	čtvrtý „odstavec“ třetí „kapitoly“, která je dítětem aktuálního uzlu
kapitola//odstavec	všechny elementy „odstavec“, které jsou potomky „kapitoly“, která je dítětem aktuálního uzlu; dvě lomítka tedy znamenají, že nezáleží na hloubce vnoření
//kapitola	všechny elementy kapitola, které potomky kořenového

	uzlu; vybere tedy všechny „kapitoly“, které jsou ve stejném dokumentu, jako aktuální uzel
//kapitola/odstavec	všechny elementy „odstavec“, které jsou dětmi „kapitol“, nacházejících se ve stejném dokumentu, jako aktuální uzel
.	aktuální uzel
./odstavec	všechny odstavec, které jsou dětmi aktuálního uzlu
..	rodiče aktuálního uzlu
@*	všechny atributy aktuálního uzlu
@typ	atribut „typ“ aktuálního uzlu
kapitola[@typ=prakticka]	všechny kapitoly, jejichž „typ=prakticka“ a jsou dětmi aktuálního uzlu
kapitola[@typ=prakticka][3]	třetí element „kapitola“, který má „typ=prakticka“ a je dítětem aktuálního uzlu
kapitola[3][@typ=prakticka]	třetí kapitolu, která je dítětem aktuálního uzlu, pokud má „typ=prakticka“
Kapitola[nadpis=Závěr]	element „kapitola“, který je dítětem aktuálního uzlu a jehož dítě „nadpis“ obsahuje text „Závěr“

Tabulka č. 1- Ukázky XPath

Další možnosti a specifikace XPath lze nalézt na webu konsorcia W3C (<http://www.w3.org/TR/xpath/>).

3.7 XLink

XLink, XML Linking Language, má původ stejně jako ostatní popisované standardy u konsorcia W3C.

Jak už název napovídá, XLink je nástroj použitelný pro tvorbu odkazů mezi XML dokumenty. Oproti obyčejným odkazům známým z HTML však přináší ještě tyto možnosti:

- možnost vytvořit odkazy mezi více, než dvěma zdroji
- možnost vytvoření odkazů uložených mimo odkazované dokumenty
- možnost přidat k odkazu metadata

Aby bylo možné XLink sadu značek v dokumentu používat, je třeba pro něj vytvořit jmenný prostor. URI adresa tohoto prostoru je „<http://www.w3.org/1999/xlink>“.

3.7.1 Jednoduché odkazy

Jednoduché odkazy jsou nejčastěji používané, jedná se o ekvivalent odkazů známých z HTML. Jednoduché odkazy se zapisují pomocí elementu „simple“.

```
<text>Další informace jsou k dispozici na <xlink:simple  
xlink:href=„http://www.w3.org/TR/xlink/“>webu </xlink:simple>.</text>
```

Cílovou adresu určuje atribut „href“. Tento příklad spojuje text „webu“ s webovou stránkou „<http://www.w3.org/TR/xlink>“. Kromě atributu „href“ lze u jednoduchých odkazů používat ještě další atributy:

- title- umožňuje vložit popis odkazu, obdoba „alt“ v HTML
- role- určuje funkci odkazu, jeho hodnoty nejsou nijak definovány
- show- říká, kde se po aktivování zobrazí cíl odkazu. Možné hodnoty jsou „embed“- zobrazí se jako součást dokumentu; „replace“- nahradí stávající dokument; „new“- dokument se otevře v novém okně
- actuate- určuje, kdy se dokument otevře; „onLoad“- automaticky při načtení dokumentu; „onRequest“- na požadavek uživatele

Odkaz nemusí být tvořen z textu, je možné jej vytvořit z jakéhokoliv elementu. V takovém případě se místo elementu s typem odkazu „<xlink:simple...“ použije typ jak atribut.

```
<autor xlink:type="simple"
      xlink:href="dp1.xml"
      title="Autor práce">Jan Dostál </autor>
```

3.7.2 Rozšířené odkazy

Narozdíl od jednoduchých odkazů mohou odkazy rozšířené spojovat více, než jen dva dokumenty. Rozšířených odkazů existují dva druhy. Inline, který musí zahrnovat alespoň jeden zdroj, který je součástí dokumentu a out-of-line, odkazující pouze na externí dokumenty.

Rozšířené odkazy se vytvářejí podobně jako jednoduché. Název elementu, respektive druh typu však musí mít hodnotu "extended". Element "extended" obvykle obsahuje další elementy, které upravují jeho funkci.. Jedná se o elementy:

- resource- v inline odkazech tento element obsahuje lokální zdroj, který je součástí odkazu
- locator- určuje jednotlivé externí zdroje
- title- stejně jako u jednoduchých odkazů je to popis pro uživatele
- arc- pomocí arc se vytvářejí jednosměrné odkazy, určuje směr odkazu

Následuje příklad:

```
<xlink:extended>
  <xlink:title>Jednotlivé práce</xlink:title>
  <xlink:locator href="dp1.xml" />
  <xlink:locator href="bp1.xml" />
</xlink:extended>
```

3.7.3 XPointer

XLink je schopný jazyk, jeho odkazy však vždy míří pouze na celé dokumenty. V mnoha případech je potřeba odkázat na určité místo v dokumentu. Xpointer dokáže místo v dokumentu adresovat s přesností jednoho znaku.

Jednoduchý odkaz na element, který má zadaný atribut „id“ lze vytvořit následovně:

```
<kapitola id="kap1">text</kapitola>
```

```
<xlink:simple href="dp1.xml#kap1">Kapitola 1</xlink:simple>
```

Při tvorbě odkazů na části, které nemají nastaveno id, se používá adresace pomocí jazyka XPath. Pro XPointer zapsaný za URL je aktuální uzel root, obsahující celý dokument. Výraz odkazující na první kapitolu tedy vypadá následovně:

```
<xlink:simple href="dp1.xml#xpointer(kapitola[1])"> První kapitola  
</xlink:simple>
```

Jazyk XPath dokáže adresovat jen jednotlivé uzly, pro potřeby Pointeru je proto rozšířen. Jsou přidány další dva druhy objektů, kterým mohou vyhovovat některé části dokumentu. Přidány jsou body a rozsahy.

Bod , čili „point“, je konkrétní místo, které lze určit s přesností na jeden znak. Nejmenší jednotkou, kterou lze adresovat tedy není element nebo atribut, ale konkrétní znak. [3]

Bod je jen pomocnou strukturou, kterou používá rozsah („range“). Rozsah je definován dvěma body a reprezentuje tu část dokumentu, která je mezi těmito dvěma body. Pomocí rozsahu tak lze adresovat libovolný úsek dokumentu, podobně jako ho lze v editoru nebo prohlížeči označit myší. [3]

Syntaxe Xpointeru na konkrétním případě vypadá následovně:

```
#start-point(//kapitola[1]) to end-point(//kapitola[2]/odstavec[3])
```

Bod lze navíc určit ještě pomocí funkce string-range, která slouží k vyhledávání řetězce v textu. Níže uvedený příklad vrátí osmý výskyt tečky v první kapitole, odkáže tedy na její devátou větu.

```
#string-range(//kapitola[1], "."[8])
```

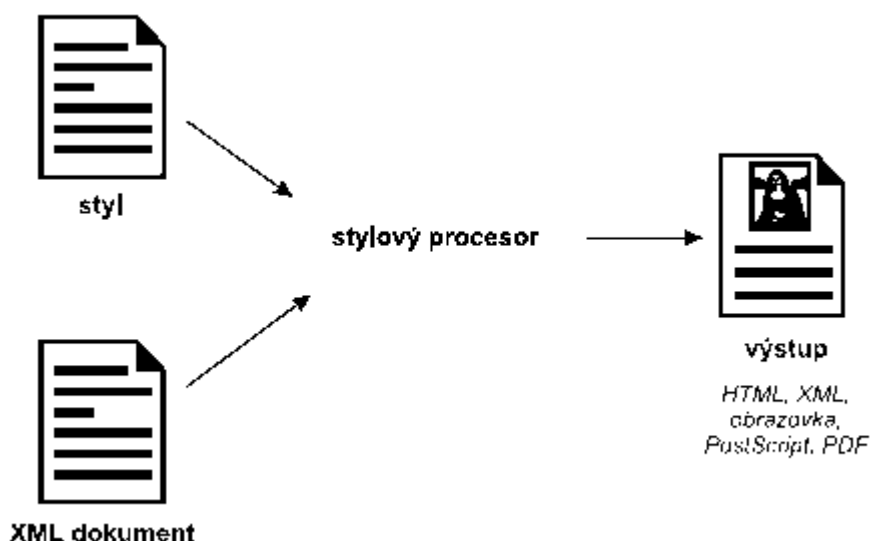
Xpointer umožňuje použít několik fragmentů za sebou, ty jsou vyhodnocovány zleva- doprava, dokud některému nevyhovuje část dokumentu.

3.8 Stylovací jazyky

Jednou ze základních myšlenek XML je oddělení informačního obsahu dokumentu od definice grafického vzhledu. Je dobře, že například vyhledávací stroj dokáže informace snadno zaindexovat, ale většina dokumentů není určena pro stroje, je určena lidem. Pro člověka je však neformátovaný XML dokument nevhodný. Je tedy potřeba z XML dokumentu vytvořit přehledné zobrazení na obrazovku, či připravit verzi pro tisk. Jak bude konkrétní výstup z XML vypadat se definuje pomocí stylových jazyků.

Ve stylovacích jazycích se tvoří styl, čili definice vzhledu jednotlivých elementů. Vytvořený styl lze poté aplikovat na několik dokumentů. Pokud je potřeba změnit některý atribut zobrazení, změní se pouze na jednom místě- v souboru se stylem. Změna se ovšem projeví při zobrazování všech stylem formátovaných dokumentů. Mimoto lze také na jeden dokument aplikovat několik stylů, čímž vytvořit několik různých výstupů.

Zpracování dokumentů na základě připojeného stylu vykonává program, kterému se říká stylový procesor. Dnes je stylový procesor součástí například každého běžně používaného webového prohlížeče.



Obrázek č. 2- Zpracování XSL

3.8.1 Připojení stylu k dokumentu

K připojení stylu se používá deklarace umístěná hned pod hlavičku XML dokumentu. Deklarace má tvar:

```
<?xml-stylesheet href="URI_stylu" type="typ_stylu"?>
```

„URI_stylu“ označuje cestu k souboru se stylem, jako „typ_stylu“ se uvádí MIME typ použitého stylového jazyka. Například pro XSL se jedná o typ „text/xsl“.

Připojení stylu tedy může vypadat například takto:

```
<?xml version="1.0" encoding="windows-1250"?>
<?xml-stylesheet href="xsl/html.xsl" type="text/xsl"?>
<prace></prace>
```

K dokumentu lze připojit několik stylů, sám uživatel si poté může vybrat, který použije. Použijí se navíc atributy „alternate“ a atribut „title“ s popisem stylu.

Ukázka použití:

```
<?xml version="1.0" encoding="windows-1250"?>
<?xml-stylesheet href="xsl/htmlcz.xsl" type="text/xsl"
alternate="yes" title="Česká verze"?>
<?xml-stylesheet href="xsl/htmlen.xsl" type="text/xsl"
alternate="yes" title="Anglická verze"?>
<prace></prace>
```

Je možné připojit i atribut „media“, pomocí kterého se určuje, pro jaký výstup je styl určen. Přípustné hodnoty atributu „media“ jsou:

- all- styl vhodný pro všechna zařízení
- screen- obrazovka počítače
- print- tiskárna
- projection- projektor
- handheld- handheld, kapesní počítač, malé rozlišení
- tv- televizní obrazovka
- tty terminal- obrazovka zobrazující pouze text, bez možnosti grafiky
- braille- hmatová čtečka Braillova písma
- aural- hlasový syntetizátor

3.8.2 XSL

XSL, eXtensible Stylesheet Language, je rodina stylových jazyků popisujících, jak se mají XML dokumenty formátovat a transformovat.

Pod standard XSL patří:

- XSLT- XML jazyk sloužící k transformaci dokumentů
- XSL:FO- formátovací objekty, jazyk určený k formátování XML dokumentů

3.8.2.1 XSLT

Jazyk XSLT, eXtensible Stylesheet Language Transformations, slouží k transformování vstupního XML dokumentu. Lze použít například k transformování do XML dokumentu s jinou strukturou, HTML stránky, textového dokumentu, či do výstupu XSL:FO. Transformace probíhá pomocí XSLT procesoru, k jejímu provedení je potřeba vstupní XML soubor a soubor XSL obsahující vzorce pro transformaci.

Základem jazyka XSLT jsou šablony. Každá šablona určuje, na jakou část vstupního XML dokumentu bude použita a jakou transformaci provede.

Obecný zápis vypadá:

```
<xsl:template match="XPath_výraz">  
vystup  
</xsl:template>
```

„XPath_výraz“ vybírá část, která se bude transformovat, „vystup“ značí výstup transformace.

Zápis používá jmenný prostor „xsl“, proto musí být všechny šablony v daném prostoru.

```
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform">  
šablony  
</xsl:stylesheet>
```


Element „apply-templates“ XSLT procesoru říká, aby na obsah vybraného elementu aplikoval další šablony. Jejich výstup je pak vložen na místo apply-templates. Jelikož mají procesory v sobě šablonu pro zpracování textu, při použití apply-templates bude vždy vypsán textový obsah elementu.

Následuje příklad, který transformuje elementy autor na tučný text, přidá navíc text „Autor práce:“.

```
<práce><autor>Jan Dostál</autor></práce>
```

```
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform">  
<xsl:template match="autor">  
  Autor práce: <b><xsl:apply-templates/></b>  
</xsl:template>  
</xsl:stylesheet>
```

Výstup šablony by vypadal takto:

```
Autor práce: <b>Jan Dostál</b>
```

3.8.2.1.1 Řazení

XSLT obsahuje několik použitelných funkcí, například řazení. Uzly, které má XSLT šablona zpracovávat lze před zpracováním seřadit. Kritéria řazení se určují pomocí elementu xsl:sort. Způsob řazení se ovládá ještě pomocí několika atributů. Těmito atributy jsou:

- select- Xpath výraz, který vybere hodnoty, podle kterých se bude řadit
- data-type- implicitně je nastaven jako “text”, pokud se mají uzly řadit jako čísla, změní se na “number”
- order- vzestupné (“ascending”), či sestupné (“descenting”) řazení
- lang- kód jazyka použitého při porovnávání

case-order- zda se mají velká písmena řadit před malá (“upper-first”), nebo naopak (“lower-first”).

Při použití více elementů „xsl:sort“ se uzly seřadí nejprve podle prvního, pokud se některé uzly shodují, tak poté i podle druhého.

Zápis tedy vypadá například takto:

```
<xsl:template match="dokument">
  <xsl:for-each select="prace">
    <xsl:sort select="typ"/>
    <xsl:sort select="autor"/>
    <xsl:value-of select="typ"/>
    <xsl:value-of select="autor"/>
    <xsl:value-of select="nazev"/>
  </xsl:for-each>
</xsl:template>
```

Příklad provede nejprve seřazení prací v dokumentu podle typu, posléze pak podle autora.

XSLT nabízí nepřehledné množství možností. Jejich popis by stačil na samostatnou práci. Některé další možnosti jsou vidět v kapitole 5, která se zabývá tvorbou praktického projektu. Kompletní specifikace tohoto standardu jsou dostupné na adrese <http://www.w3.org/TR/xslt>.

3.8.2.2 XSL:FO

XSL:FO je spojeným prezentačním jazykem. Je součástí standardizovaného jazyka XSL. Dokument XSL:FO v sobě obsahuje veškerá data dokumentu, jak informace o formátování, tak vlastní formátované elementy.

Základní myšlenkou je, že netvoříme FO dokument, ale XML dokumenty a XSL soubory, pomocí nichž se XML dokument na FO přetransformuje. Existují však i editory, pomocí nichž se v grafickém prostředí tvoří přímo FO dokument.

3.8.2.2.1 Struktura XSL:FO dokumentu

XSL:FO používá jmenný prostor a používá kořenová element root. Kořenový element obsahuje dva další základní elementy. Těmito elementy jsou „fo:layout-master-set” a „fo:page-sequence”. Fo:layout-master-set umožňuje definovat layout stránky, čili její velikost, okraje a podobně. Element „fo:page-sequence“ poté již zastupuje obsah jednotlivých stránek. Element „fo:page-sequence“ může obsahovat ještě elementy „fo:static-content“, používané pro opakující se části stránek, typicky záhlaví nebo zápatí. Objekty, které mají být zobrazeny v hlavní části dokumentu se poté vkládají dovnitř elementu „flow“.

Obecný XSL soubor sloužící k vytvoření XSL:FO dokumentu by měl mít takovou strukturu:

```
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
                xmlns:fo="http://www.w3.org/1999/XSL/Format">
  <xsl:template match="/">
    <fo:root>
      <fo:layout-master-set>
        zde by se nacházelo nastavení layoutu stránky
      </fo:layout-master-set>
      <fo:page-sequence master-reference="my-page">
        <fo:static-content flow-name="xsl-region-after">
          místo pro zápatí
        </fo:static-content>
        <fo:flow flow-name="xsl-region-body">
          místo pro obsah stránek
        </fo:flow>
      </fo:page-sequence>
    </fo:root>
  </xsl:template>
```

Pro přesné naformátování stránky umožňuje XSL:FO použít mnoho druhů objektů, které obsahují nepřeborné množství atributů. Jejich seznam a vlastnosti jsou k dispozici na stránce <http://www.w3.org/TR/xsl/#fo-section>.

4. OpenDocument Format

OpenDocument, celým názvem celým názvem OASIS Open Document Format for Office Applications, je otevřený souborový formát určený pro ukládání a výměnu kancelářských dokumentů. ODF zahrnuje textové dokumenty prezentace, tabulky, grafy a databáze [13].

Druhy dokumentu i šablon a k nim uvedené koncovky souborů jsou v tabulce č. 2.

Dokument	Přípona	Přípona šablony
Textový dokument	.odt	.ott
Tabulka	.ods	.ots
Prezentace	.odp	.otp
Grafika	.odg	.otg
Graf	.odc	.otc
Matematický výraz	.odf	.otf
Databáze	.odb	
Obrázek	.odi	.oti
Hlavní textový dokument	.odm	
Šablona webové stránky		.oth

Tabulka č. 2- Druhy souborů

4.1 OASIS

OASIS (Organization for the Advancement of Structured Information Standards) je nezisková organizace, která řídí vývoj, konvergenci a přijímání otevřených standardů pro globální informační společnost. Byla založena v roce 1993, nyní má více než 600 spolupracujících organizací ve více než 100 zemích. Vedení je vybíráno demokratickou cestou s dvouletým mandátem.

Kroky OASIS jsou transparentní, všechny specifikace navržené ke standardizaci jsou podrobeny veřejné diskuzi

Konsorcium produkuje více webových standardů, než jakákoliv jiná organizace. Produkuje také standardy pro bezpečnost, e-byznys a státní sektor.

Výsledkem činnosti OASIS je již mnoho kvalitních otevřených standardů, architektur a služeb pro práci s daty. Kromě OpenDocument jsou to například ebXML (architektura pro elektronickou komunikaci mezi firmami), UDDI (služba pro zveřejňování a vyhledávání podnikových informací), SAML (standard pro výměnu autentizačních a autorizačních dat mezi bezpečnostními doménami) WS-Security (komunikační protokol pro bezpečné webové služby) nebo DocBook (formát pro technickou dokumentaci). Kromě práce na standardech vznikají v rámci OASIS také různé specializované projekty, jako je LegalXML (datové standardy pro právo a soudnictví), eGovernment (elektronická veřejná správa), Open CSA (otevřená architektura kompozitních služeb), SOA-RM (referenční model architektury orientované na služby) a mnoho dalších [13].

4.2 Historie

Standart ODF byl vyvinut sdružením OASIS. Jednotlivé milníky vývoje jsou uvedeny v tabulce č. 3.

Datum	Událost
1999	Ve StarDivision začíná vývoj formátu založeném na standartu XML. Spouštěcím mechanismem je potřeba jsou omezení předchozích binárních formátů a podpory Unicode. Cílem je vytvořit otevřený formát souboru, který bude moci být využíván také dalšími společnostmi.
srpen 1999	Sun Microsystems, Inc . získávají StarDivision
13.říjen 2000	Sun Microsystems, Inc., uvolňuje nově vzniklému projektu OpenOffice.org zdrojový kód StarOffice.
18.říjen 2000	Je přizvána komunita kolem XML, aby vytvořila jasné specifikace otevřeného formátu OpenOffice.Org XML.
2002	Do specifikace OpenOffice.Org XML přidány definice pro čínský, japonský a korejský jazyk. Dále pak pro jazyky textové.
2002	Začíná spolupráce s projektem KOffice

16. prosinec 2002	První konference technické komise OASIS Open Office
květen 2003	Uvolnění OpenOffice.Org 1.0 a StarOffice 6, oba produkty využívají jako standartní formát souborů OpenOffice.Org XML.
Srpen 2003	Koffice začíná jako alianční formát používat ODF.
2003 / 2004	<p>Původní specifikace souborového formátu OpenOffice.Org XML je upravena, zohledňuje poslední trendy v rozvoji XML a kancelářských aplikací, například:</p> <ul style="list-style-type: none"> -OASIS upravuje pravidla po zavedení jmenných prostorů v XML - přechod z XML DTD do Relax-NG součástí schématu -zlepšení schémat pro lepší podporu validace dokumentů -přizpůsobení schématu novým verzím standartů -přizpůsobené pro další kancelářské aplikace (Koffice) -přizpůsobení pro nové verze aplikací (OpenOffice.Org 2.0) -odstranění nesrovnalostí ve specifikacích -opravy chyb
prosinec 2004	Schválena změna názvu z „OASIS Open Office Specification“ na „OASIS Open Alliance Format for Office Applications (OpenDocument)“
leden 2005	TC je přejmenovaný na „OASIS Open format alliance for Office Applications (OpenDocument) TC.“
únor 2005	Výbor schválil třetí návrh specifikace formátu souborů , včetně zhodnocení připomínek veřejnosti.
květen 2005	OpenDocument Format je schválený jako standart OASIS
září 2005	Sun Microsystems vydávají StarOffice 8 s podporou ODF.
září 2005	ODF je předložený Mezinárodní organizaci pro normalizaci (ISO).
září 2005	INDT (výzkumná skupina patřící alianci) přispívá ODF filtry pro Abiword a Gnumeric.
říjen 2005	Vydání OpenOffice.org 2.0 s podporou ODF
říjen 2005	<p>Sun vydává patent:</p> <p>OpenDocument není zatížen žádnými právními břemeny. (http://xml.coverpages.org/ni2005-10-04-a.html)</p>

prosinec 2005	Textmaker 2006 s podporou ODF
leden 2006	IBM vydává IBM Workplace s podporou ODF
březen 2006	Založena ODF Aliance. Jejím úkolem je získat podporu ODF ve veřejném sektoru. 35 zakládajících členů
duben 2006	Koffice 1.5, standardní formát ODF
květen 2006	ISO schvaluje ODF jako ISO/IEC 26300
červen 2006	ODF aliance má již více než 200 členů. Například BBC, Corel, EDS, EMC, IBM, Novell, Red Hat, Oracle, software AG, Sun M.
září 2006	ODF 1.0 druhá edice, dokončení redakčních úprav stanovených v procesu zhodnocení ISO
listopad 2006	ODF 1.1 schváleno jako specifikace výboru OASIS. ODF Aliance má nyní více, než 300 členů.

Tabulka č. 3- Historie ODF

4.3 Proč používat ODF

Dokumenty jsou jako životní krev moderních firem. Firmy užívají dokumenty k tomu, aby ukládaly rozhodující informace, koordinovaly činnosti, měřily výsledky a mohly komunikovat napříč odděleními, s dodavateli i zákazníky.

Stále více dokumentů se přesouvá z papíru do elektronických formulářů. Aby byla možnost přizpůsobit se měnící technologii, firmy potřebují jistotu, že záznamy budou přístupné stejně tak dnes, jako v budoucnu.

OpenDocument Format reaguje na tento problém a přináší firmám skutečnou kontrolu nad jejich dokumenty. Firmy využíváním aplikací podporujících ODF získávají větší efektivitu, flexibilitu a technologii vedoucí ke zvýšení schopností

4.3.1 OpenDocument je standard

OpenDocument je standardizován jako norma ISO/IEC 26300:2006. To znamená, že formát splnil přísné podmínky standardizačního procesu a byl zařazen mezi celosvětově uznávané standardy. Jedním z důležitých důvodů je i to, že pro uložení a interpretaci dat používá specifikace založené na dalších standardizovaných formátech [7].

OpenDocument byl doporučen Evropskou komisí jak formát vhodný pro komunikaci orgánů veřejné správy s občany i uvnitř samotných orgánů. Toto doporučení se opírá především o technologickou neutralitu formátu, otevřenost specifikace a jednoznačnou právní čistotu (neexistující licenční a patentové otazníky) [7].

OpenDocument byl doporučen Ministerstvem informatiky České republiky jako formát vhodný pro komunikaci orgánů veřejné správy s občany. Formát totiž umožňuje beze zbytku splnit povinnosti uložené § 4, odst. 2, zákona č. 106/1999 Sb. (o svobodném přístupu k informacím) [7].

OpenDocument format je standardizován také jako americký standard americký standard INCITS/ISO/IEC 26300-2007.

4.3.2 Otevřený standard

Vývoj, udržování a řízení otevřeného standardu je přesný manažerský proces. Jsou předloženy jednotlivé požadavky, seřazeny podle důležitosti a posléze zahrnuty do standardu.

Tento proces je demokratický, přispívat mohou členové, koncoví uživatelé, později i standardizační organizace. Díky tomuto přístupu nemůže jedna společnost měnit formát a licenční podmínky bez souhlasu ostatních.

OpenDocument Format byl vyvinut velkým množstvím organizací, je to otevřený standard, což znamená, že specifikace formátu je zdokumentovaná a volně přístupná. Kdokoliv, včetně koncových uživatelů může vidět specifikace.

Formát OpenDocument je jednoduchý. Implementace standardu do aplikace, která bude plně podporovat všechny potřebné funkce a vlastnosti formátu OpenDocument, není složité. To významně usnadňuje použití formátu pro výstupy z informačních systémů, pro výměnu dat mezi různými systémy a pro datové transformace. Navíc lze kdykoli později vytvořit aplikaci, která bude zpracovávat dokumenty vytvořené v minulosti. To odstraňuje bariéry, podporuje zdravou konkurenci, zvyšuje efektivnost nákladů a umožňuje inovační alternativy.

Formát ODF není zatížen žádnými právními břemeny. Žádná jeho část nespadá pod nějakou licenci, či patent, díky čemuž může být použit libovolnou aplikací bez omezení a strachu v podobě nevyjasněných nároků, či poplatků.

Standard není spojen s žádnou aplikací, dodavatelem softwaru, či platformou. To přináší pro uživatele možnost volby mezi aplikacemi, pro standard poté jeho dlouho životnost. Podpora standardu více aplikacemi samozřejmě zvyšuje možnosti jejich součinnosti.

4.3.3 Podpora

Na podporu formátu ODF byla 3. března 2006 zřízena ODF Aliance, která má nyní několik stovek členů. Namátkou jsou to například BBC, Corel, EDS, EMC, Google Inc, IBM, Novell, Red Hat, Oracle, software AG, Software602, Sun Microsystems

Existuje globální potřeba pro dlouhodobý přístup k dokumentům a záznamům pomocí různých aplikací, což demonstruje rostoucí propagace i užívání ODF.

Mnoho vlád, napříč celým Světem, rozhodlo o podpoře ODF. Jedná se například o Belgii, Brazílii, Dánsko, Francii, Německo, Nizozemí, a Norsko. Ostatně ODF doporučuje i Evropská komise a Ministerstvo informatiky ČR.

Firmy reagují na doporučení vlád a potřeby zákazníků, do svých produktů podporu ODF implementují.

V současnosti je na trhu mnoho různorodých aplikací využívajících ODF. Komerční i open source. Jsou jimi například OpenOffice.org, KOffice, StarOffice, Corel WordPerfect, Lotus Symphony či webové aplikace Google Docs a Zoho.

Vzhledem k jednoduchosti implementace se jejich počet rychle zvyšuje. Protože formát vznikl za účasti světových leaderů v oblasti podnikových informačních systémů, jsou tyto společnosti schopny bezproblémově použít tento formát pro vstup a výstup dat, pro dokumentové toky, pro mezisystémová rozhraní a další účely.

4.3.4 Státní správa

Při používání ve státní správě nabízí ODF tyto hlavní výhody:

Přístup k informacím

Při používání uzavřeného formátu mohou součásti státní správy v budoucnu ztratit schopnost otevřít, či modifikovat archivované dokumenty.

ODF, jako otevřený standard dokáže zajistit, že uložené dokumenty přístupné i v budoucnu. Přístup k uloženým informacím by v oblasti státní správy neměl být nikdy závislý na technologii jediného dodavatele.

Požadavky zákona 106/1999 Sb.

Zákon č. 106/1999 Sb. (o svobodném přístupu k informacím), v aktuálním znění, obsahuje v § 4, odst. 2 povinnost poskytovat zveřejňované informace ve formátech, jejichž specifikace je volně dostupná a jejich použití uživatelem není omezováno. Ministerstvo informatiky ČR vydalo doporučený seznam formátů, které tyto podmínky splňují. Pro textové dokumenty, tabulky, grafy a prezentace je doporučeným formátem OpenDocument[7].

Součinnost

ODF pomáhá oddělit dokument od aplikace, která ho vytvořila. Tento dokument pak může být bezproblémově zpracován jinými aplikacemi, to vše bez patentového nebo jiného omezení.

Konkurence

Jelikož ODF je opravdu otevřený standard, dává to softwarovým firmám vyrovnané možnost na trhu soutěžit pomocí vlastností a ceny.

Možnost volby

Státní správa často nemá na výběr. Její informační strategie musí jít cestou aktualizací stávajících produktů i když již nejsou cenově, ani funkčně zajímavé. ODF při výběru budoucí strategie a změně dodavatele netvoří zbytečné bariéry.

Cena

ODF je cenově dostupný, jelikož na trhu funguje mezi aplikacemi zdravá konkurence. Občané nebudou nuceni za přístup k informacím platit, existují i bezplatná řešení

Inovace

Ve formě ODF byl poskytnut na platformě nezávislý standard, na kterém může každá společnost tvořit nové aplikace. Při dodržení základního standardu budou dokumenty přístupné i po přidání nových funkcí.

Záchrana kulturního dědictví

Stále více dokumentů potenciálně historického významu je tvořeno a ukládáno v digitální formě. Je podstatné, aby státní správa udržela schopnost tyto dokumenty zpřístupňovat nejen nyní, ale i pro budoucí generace. ODF je na trhu aktuálně jediný otevřený formát, který to umožňuje.

Naléhavé případy

Potřeba otevřených standardů je přesvědčivě vidět i v kontextu naléhavých případů.

V roce 2004 zasáhlo ničivé tsunami Thajsko. Jeho vláda, domácí a mezinárodní agentury nebyly schopny zabezpečit přístup k informacím, jež byly klíčové pro záchranné operace. To vše, protože používaly různé formáty dokumentů. Veřejný přístup k informacím by nikdy neměl být omezen na uživatele programů jediné společnosti.

4.3.5 Firemní prostředí

Používání formátu OpenDocument může ve firemním prostředí přinést významnou úsporu nákladů a mnoho dalších výhod. Hlavní výhody:

Standardizace

Protože se formát OpenDocument stal normou ISO/IEC, jednak to potvrdilo jeho kvalitní návrh a vhodnost pro univerzální použití, ale současně to přináší výhody pro

použití v procesech kontrolovaných v rámci systémů řízení jakosti s certifikací podle norem ze souboru ISO 9000 a dalších podobných předpisů [7].

Otevřenost a transparentnost formátu

Formát OpenDocument je otevřený a plně specifikovaný. S dokumenty uloženými ve formátu ODF půjde pracovat kdykoli v budoucnu a to i pokud nebude z jakéhokoliv důvodu k dispozici aplikace, která je vytvořila. Na základě standardu je totiž možné v případě potřeby vytvořit aplikaci jinou. To je podstatný rozdíl oproti uzavřeným formátům, které může být problém později zpracovávat.

Univerzálnost

Vytvořené dokumenty nejsou pevně spjaté s jakoukoliv platformou či programem.

Součinnost

Formát OpenDocument se hodí pro výměnu dat mezi zcela odlišnými aplikacemi nebo prostředími. V toku dokumentů může figurovat podnikový informační systém, kancelářské aplikace, mobilní a webové aplikace, rešeršní systémy a mnoho dalšího.

Právní jistota

Na rozdíl od některých jiných formátů, dává OpenDocument právní jistotu pro firmu i její klienty. Nehrozí totiž, že by někdo mohl uplatňovat licenční nebo patentové nároky. Přestože softwarové patenty nejsou v České republice a v celé Evropské unii legislativně uznány, například ve Spojených státech amerických nebo Japonsku jsou uplatňovány a při použití formátů s nejasným právním stavem by mohlo hrozit riziko klientům v těchto státech [7].

Cena

Implementace ODF do informačních systémů je jednodušší, a tedy i levnější, než v případě jiných srovnatelných formátů. I pořízení dalších aplikací pro práci s dokumenty bývá díky lepšímu konkurenčnímu prostředí povětšinou levnější.

OASIS

Na tvorbě formátu OpenDocument se podílela většina nejvýznamnějších světových společností zabývajících se tvorbou kancelářských aplikací, podnikových informačních systémů, operačních systémů, počítačové bezpečnosti a řady dalších oborů. To dává záruku, že vlastnosti formátu odpovídají požadavkům praktického použití. Současně to znamená, že společnosti, které se na tvorbě formátu podílely, ho budou schopny (pokud tak již neučinily) bez problémů začlenit do svých produktů a řešení [7].

4.3.6 Školství, věda a výzkum

Při použití ODF v sektoru školství, vědy a výzkumu platí povětšinou výhody z prostředí státní správy, přidávají se však i další.

Odstranění překážek

Pro efektivní vzdělávací proces je potřeba, aby studenti i učitelé měli možnost pracovat se svými dokumenty i v prostředí mimo školu. Pokud by používali formát vázaný na konkrétní softwarovou aplikaci, znamenalo by to v podstatě nutnost zakoupit licenci na tuto aplikaci. To však může být finanční, případně i technickou překážkou efektivní práce. Formát OpenDocument tuto překážku odbourává, protože již dnes existuje mnoho programů, které s ním pracují a řada z nich je k dispozici zcela zdarma [7].

Všestrannější výuka

Široká škála programů pracujících s formátem OpenDocument dává možnost používat při výuce více různých aplikací a připravit tak studenty na práci v různém prostředí. Ti jsou pak schopni mnohem pružněji přecházet mezi programy (i jiného druhu než jen pro kancelářskou práci), aniž by potřebovali zvláštní zaškolování.

Schopnost spolupráce

Mnoho významných výzkumných ústavů, univerzit, středních i základních škol používá pro ukládání dat ODF. Stejný formát pak usnadňuje spolupráci.

4.3.7 Domácnosti a malé firmy

Pro využití v domácnostech a malých firmách nabízí ODF následující hlavní výhody.

Svoboda

Závislost na jednom poskytovateli software může znamenat velký problém například při změně jeho licenční politiky, či odchodu z trhu. Při použití ODF mohou uživatelé svobodně volit mezi produkty různých firem, či volně přecházet mezi různými platformami. Nejsou přitom limitováni potřebou editace již vytvořených dokumentů.

Cena

Mnoho domácností i malých firem je nuceno šetřit každou korunu, cena pak může být hlavním faktorem ovlivňujícím výběr produktu. Řada programů pracujících s ODF je k dispozici buď zcela zdarma, nebo za přijatelnou cenu.

Komunikace s veřejnou správou

Formát ODF je v ČR kvůli splnění zákonných podmínek používán veřejnou správou. Proto je pro komunikaci s ní výhodné používat stejný formát.

5. Použití v praxi

Praktickou část diplomové práce jsem si rozdělil na tři části.

První část se věnuje tvorbě XML v té nejzákladnější podobě- pomocí poznámkového bloku, případně pomocí editorů pouze zvýrazňujících napsaný text.

Ve druhé části převádím pomocí komerčního softwaru 602XML Designer do formátu .fo tištěný formulář společnosti PRE.

Poslední, třetí část se věnuje porovnání formátu OpenDocumentFormat, respektive programu OpenOffice.org Writer s konkurencí.

5.1 XML

V této části vytvořím, za použití výše popsaných technologií, modelový příklad pro praktické využití XML. Vytvořený model bude ze vstupních XML souborů obsahujících bakalářské, respektive diplomové, práce tvořit různé výstupy. Ačkoliv bych mohl využít některý z editorů, které zřejmě usnadňují práci, využiji pouze poznámkový blok, což mi umožní mít kontrolu nad všemi částmi kódu. Poté zhodnotím klady a zápory XML, na které jsem při tvorbě narazil, případně navrhu, kam by se vývoj mohl ubírat do budoucna.

5.1.1 DTD

Jako první jsem vytvořil DTD- definici typu dokumentu. Funkce jednotlivých částí je popsána v komentářích

prace.dtd:

```
<?xml version="1.0" encoding="windows-1250"?>

<!-- Prace musi obsahovat element LOGO 0/1x, NAZEV CZ, NAZEVEN, AUTORA,
VEDOUCI nemusi byt uveden, -->
<!-- nasleduje 0/1 OBSAH a jedenkrat TEXT -->
<!ELEMENT prace (logo?, nazevcz, nazeven, autor, vedouci? , obsah?,
text)>

    <!-- Seznam atributu elementu PRACE, typ je vyžadovan, může nabývat
hodnot BR/DP -->
    <!-- Název souboru je nutné zadat, klíčová slova být nemusí -->
    <!ATTLIST prace typ (BP | DP) #REQUIRED
        nazevs NMTOKEN #REQUIRED
        key NMTOKENS #IMPLIED
        keyen NMTOKENS #IMPLIED>

    <!-- Element autor může obsahovat text -->
    <!-- Uživatelské jméno autora je vyžadováno, email volitelný -->
```

```

<!ELEMENT autor (#PCDATA)>
  <!ATTLIST autor username CDATA #REQUIRED
                email CDATA #IMPLIED>

  <!-- U vedoucího můžeme zadat email -->
<!ELEMENT vedouci (#PCDATA)>
  <!ATTLIST vedouci email CDATA #IMPLIED>

<!ELEMENT nazevcz (#PCDATA)>
<!ELEMENT nazeven (#PCDATA)>

<!ELEMENT obsah (odkaz*)>

  <!-- Element odkaz musí mít zadán atribut CIL -->
<!ELEMENT odkaz (#PCDATA)>
  <!ATTLIST odkaz cil IDREF #REQUIRED>

  <!-- Element TEXT musí obsahovat Uvod, poté je možné mít několik
KAPITOL, musí následovat ZAVER a dále je ještě -->
  <!-- možné mít další kapitoly -->
<!ELEMENT text (uvod, kapitola*, zaver, kapitola*)>

  <!-- UVOD obsahuje NADPIS1 a dále alespoň jeden element P. ID
vyžadováno -->
<!ELEMENT uvod (nadpis1, p+)>
  <!ATTLIST uvod id ID #REQUIRED>

<!ELEMENT zaver (nadpis1, p+)>
  <!ATTLIST zaver id ID #REQUIRED>

  <!-- KAPITOLA obsahuje některý ze tří druhů nadpisů, dále libovolné
množství IMG a P. ID vyžadováno -->
  <!ELEMENT kapitola ((nadpis1 | nadpis2 | nadpis3), (p | img), (p* |
img*)* ) >
  <!ATTLIST kapitola id ID #REQUIRED>

  <!-- Prázdný element LOGO, musí obsahovat atribut URL -->
<!ELEMENT logo EMPTY>
  <!ATTLIST logo url CDATA #REQUIRED>

<!ELEMENT p (#PCDATA)>

  <!-- Prázdný element IMG. ZAROVNANI můžeme nastavit z hodnot
left/center/right, implicitně nastaveno LEFT -->
  <!-- Další atributy jsou povinné URL a ID, dále pak volitelný ALT -->
<!ELEMENT img EMPTY>
  <!ATTLIST img zarovnaní ( left | center | right) "left"
                id ID #REQUIRED
                alt CDATA #IMPLIED
                url CDATA #REQUIRED>

  <!-- Nadpisy obsahují text -->
<!ELEMENT nadpis1 (#PCDATA) >
<!ELEMENT nadpis2 (#PCDATA) >
<!ELEMENT nadpis3 (#PCDATA) >

```


Druhý vytvořený DTD seznam.dtd (přiložen na CD) obsahuje pravidla, která musí dodržovat soubor seznam.xml.

5.1.2 Zkušební XML

Podle pravidel obsažených v souboru práce.dtd jsem vytvořil dva zkušební soubory. Bp1.xml, jehož obsah je možné vidět v přílohách, či na přiloženém CD a dp1.xml, jehož obsah následuje.

dp1.xml:

```
<?xml version="1.0" encoding="windows-1250"?>
<!-- <?xml-stylesheet href="xsl/html.xsl" type="text/xsl"? -->

<!-- Pripojeny DTD soubor -->
<!DOCTYPE prace SYSTEM "dtd/prace.dtd">
<prace typ="DP" key="XML DTD XSLT XPath XLink XPointer Filler602
602Designer OpenDocument" keyen="XML DTD XSLT XPath XLink XPointer
Filler602 602Designer OpenDocument " nazevs="dp1">
  <logo url="img\LOGO_CZU.jpg"/>
  <nazevcz>XML a OpenDocument Format standardy a jejich použití v
praxi</nazevcz>
  <nazeven>XML and OpenDocument Format standards and their use in
practice</nazeven>
  <autor username="xdosj301" email="dostyx@t-email.cz">Jan
Dostál</autor>
  <vedouci>Ing. Čestmír Halbich CSc.</vedouci>
  <obsah>
    <odkaz cil="kap5">Kapitola 5</odkaz>
  </obsah>
  <text>
    <uvod id="kap1">
      <nadpis1>1. Úvod</nadpis1>
      <p>Zde se bude nacházet úvod</p>
    </uvod>
    <kapitola id="kap2">
      <nadpis1>2. Cíl práce a metodika</nadpis1>
      <p></p>
    </kapitola>

    <kapitola id="kap3">
      <nadpis1>3. XML</nadpis1>
      <p>Text kapitoly XML</p>
      <img url="img/devil.jpg" zarovnnani="center" alt="Obrazek
dablika" id="obr1"/>
    </kapitola>

    <kapitola id="kap4">
      <nadpis1>4. OpenDocument</nadpis1>
      <p>dee</p>
    </kapitola>

    <kapitola id="kap5">
```

```
        <nadpis1>5. Použití v praxi</nadpis1>
        <p>fggh</p>
</kapitola>

<kapitola id="kap5.1">
    <nadpis2>5.1 XML</nadpis2>
    <p></p>
</kapitola>

<kapitola id="kap5.2">
    <nadpis2>5.2 602 Designer</nadpis2>
    <p></p>
</kapitola>

<kapitola id="kap5.3">
    <nadpis2>5.3 OpenDocument</nadpis2>
    <p></p>
</kapitola>

<zaver id="kapzaver">
    <nadpis1>6. Závěr</nadpis1>
    <p>A zde závěr.</p>
    <p>Závěr nemá jen jeden odstavec.</p>
    <p>Má dokonce tři.</p>
</zaver>

<kapitola id="kap7">
    <nadpis1>7. Seznam použitých zdrojů</nadpis1>
    <p></p>
</kapitola>

<kapitola id="kap8">
    <nadpis1>8. Přílohy</nadpis1>
    <p></p>
</kapitola>

</text>
</prace>
```

Ručně jsem vytvořil ještě soubor seznam.xml, který obsahuje seznam odevzdaných prací.

5.1.3 XSLT

Dalším krokem modelového příkladu bylo vytvoření XSLT stylsheetů, sloužících k transformování vstupních souborů na požadovaný výstup. K zpracování byl využit procesor SAXON-HE 9.3.0.4.

5.1.3.1 Html výstupy

Zde je obsah XSLT souboru html.xml, který transformuje vstupní xml soubory do výstupního formátu html.

html.xml:

```
<?xml version="1.0" encoding="windows-1250"?>
<!-- Transformuje XML na HTML -->
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  exclude-result-prefixes="xs"
  version="2.0"
  xmlns:saxon="http://icl.com/saxon"
  extension-element-prefixes="saxon">

  <!-- Musime procesoru SAXON rucne rici, jake ma pouzit kodovani. Typ
výstupu. HTML -->
  <xsl:output method="html" encoding="windows-1250"
    saxon:character-representation="native"/>

  <!-- Vytvori strukturu HTML stranky -->
  <xsl:template match="/">
    <html>
      <head>
        <title>
          <xsl:value-of select= "/prace/@typ"/>
          <xsl:value-of select= "/prace/nazevcz"/>
        </title>
      </head>
      <body>
        <xsl:apply-templates/>
      </body>
    </html>
  </xsl:template>

  <!-- Transformace elementu NAZEV CZ -->
  <xsl:template match="nazevcz">
    <hr></hr>
    <h1><xsl:apply-templates/></h1>
  </xsl:template>

  <xsl:template match="nazeven">
  </xsl:template>

  <xsl:template match="odkaz">
  </xsl:template>
```

```

<xsl:template match="uvod">
  <!-- U UVODu vytváří návěští pro html odkazy -->
  <a><xsl:attribute name="name"><xsl:value-of
select="@id"/></xsl:attribute></a>
  <xsl:apply-templates />
</xsl:template>

<xsl:template match="zaver">
  <!-- U ZAVERu vytváří návěští pro html odkazy -->
  <a>
    <xsl:attribute name="name"><xsl:value-of
select="@id"/></xsl:attribute>
  </a><xsl:apply-templates />
</xsl:template>

<xsl:template match="kapitola">
  <!-- U KAPITOLY vytváří návěští pro html odkazy -->
  <a>
    <xsl:attribute name="name"><xsl:value-of
select="@id"/></xsl:attribute>
  </a><xsl:apply-templates />
</xsl:template>

<xsl:template match="autor">
  <!-- AUTORa transformuje na odkaz typu mailto -->
  Autor práce: <b><a>
    <xsl:attribute name="href">mailto:<xsl:value-of
select="@email"/></xsl:attribute>
    <xsl:apply-templates />
  </a></b><br/>
</xsl:template>

<!-- Transformace pro vlozeni obrazku -->
<xsl:template match="img">
  <img>
    <xsl:attribute name="src"><xsl:value-of
select="@url"/></xsl:attribute>
  </img><br/><xsl:value-of select="@alt"/><br/>
  <xsl:apply-templates />
</xsl:template>

<xsl:template match="vedouci">
  Vedoucí práce: <xsl:apply-templates/>
</xsl:template>

<xsl:template match="nadpis1">
  <h2><xsl:apply-templates/></h2>
</xsl:template>
<xsl:template match="nadpis2">
  <h3><xsl:apply-templates/></h3>
</xsl:template>
<xsl:template match="nadpis3">
  <h3><xsl:apply-templates/></h3>
</xsl:template>
<!-- První slovo odstavce bude o 10x odsazeno -->
<xsl:template match="p">
  <p style="padding-left:10px"><xsl:apply-templates/></p>
</xsl:template>
</xsl:stylesheet>

```

Na obrázku č. 3 je vidět, jak se v prohlížeči Chrome zobrazí výstupní soubor dp1.xml.

XML a OpenDocument Format standardy a jejich použití v praxi

Autor práce: [Jan Dostál](#)
Vedoucí práce: Ing. Čestmír Halbich CSc.


1. Úvod

Zde se bude nacházet úvod

2. Cíl práce a metodika

3. XML

Text kapitoly XML



Obrázek dablíka

4. OpenDocument

dee

5. Použití v praxi

fgghh

5.1 XML

5.2 602 Designer

5.3 OpenDocument

6. Závěr

A zde závěr.

Závěr nemá jen jeden odstavec.

Má dokonce tři.

7. Seznam použitých zdrojů

8. Přílohy

Obrázek č. 3- dp1.html

Obrázek č. 4 ukazuje zobrazení souboru menudp1.html vytvořeného aplikací XSLT osnovahtml.xml (viz CD). Tento styl zároveň voláním html.xml vytvoří i předchozí výstup dp1.html.



Obrázek č. 4- menudp1.html

Názvy jednotlivých kapitol slouží jako odkazy, které vedou na jednotlivé kapitoly v souboru dp1.html.

Český název	Typ práce	Autor	Vedoucí
XML a OpenDocument Format standardy a jejich použití v praxi	DP	Jan Dostál	Ing. Čestmír Halbich CSc.
Vyhledávač google	BP	Jan Dostál	Ing. Čestmír Halbich CSc.

Obrázek č. 5- Seznam.html

Obrázek č. 5 ukazuje zobrazení souboru seznam.html. Tento soubor byl vytvořen aplikací XSLT seznamhtml.xml na soubor seznam.xml.

České názvy prací slouží jako odkazy na webové stránky [menunazev.html](#).

5.1.3.2 Výstup SQL

Předchozí XSLT stylsheety tvořily výstupy ve tvaru HTML. V této krátké kapitole ukážu výstup stylu xml2sql.xsl (viz CD).

Jako vstup pro tento soubor využívám již vytvořený soubor se seznamem prací seznam.xml (viz CD).

Výstupem je poté soubor input.sql, který obsahuje sérii SQL příkazů a lze jej použít pro zadání dat do relační databáze.

Obsah souboru input.sql:

```
INSERT INTO Prace (TypP, Soubor, UN, Autor, Vedouci, NazevCZ) VALUES ('DP',  
'dp1.xml', 'xdosj301', 'Jan Dostál', 'Ing. Čestmír Halbich CSc.', 'XML a OpenDocument  
Format standardy a jejich použití v praxi');
```

```
INSERT INTO Prace (TypP, Soubor, UN, Autor, Vedouci, NazevCZ) VALUES ('BP',  
'bp1.xml', 'xdosj301', 'Jan Dostál', 'Ing. Čestmír Halbich CSc.', 'Vyhledávač google');
```

5.1.3.3 Výstup do formátu .fo

Prezentace prací na webu, případně jejich vložení do databáze je zajímavé, ovšem při tvorbě bakalářských i diplomových prací se jistě počítá hlavně s vytvořením jejich tištěné podoby.

V rešeršní části této práce již bylo řečeno, že sada značek určená pro popis vzhledu tištěných dokumentů se nazývá formátovací objekty. Tuto sadu značek jsem tedy využil k generování formátu .fo a následně i uložení do .pdf.

V prvním kroku je třeba na soubor s prací dp1.xml aplikovat stylsheety soubory.xml a fotitul.xml (oba na CD), což vygeneruje soubory souborydp1.xml a tituld1.xml.

Tituld1.xml slouží k duplikaci vstupních dat za účelem vytvoření titulní stránky a obsahu, soubor souborydp1.xml poté spojuje základní dp1.xml a tituld1.xml.

V druhém kroku aplikuji na souborydp1.xml styl fo2.xml

fo2.xsl:

```
<?xml version="1.0" encoding="windows-1250"?>
<!-- Vstupní XML soubor transformuje na FO object -->
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  exclude-result-prefixes="xs" version="2.0"
  xmlns:fo="http://www.w3.org/1999/XSL/Format">

  <xsl:template match="/">
    <fo:root>
      <!-- Layout stránky -->
      <fo:layout-master-set>
        <!-- Rozměry a okraje -->
        <fo:simple-page-master master-name="my-page"
          page-height="297mm"
          page-width="210mm"
          margin="1in">
          <!-- Nastavení vnějších okrajů stránky -->
          <fo:region-body margin-top="30mm"/>
          <fo:region-body margin-left="35mm"/>
          <fo:region-body margin-bottom="30mm"/>
          <fo:region-body margin-right="20mm"/>
          <!-- Velikost patky -->
          <fo:region-before extent="0mm"/>
          <fo:region-after extent="3mm"/>
        </fo:simple-page-master>
      </fo:layout-master-set>

      <!-- Definice obsahu stránky -->
      <fo:page-sequence master-reference="my-page">
        <!-- Vložení čísla stránek do záhlaví -->
        <fo:static-content flow-name="xsl-region-after">
          <fo:block font-size="10" text-align="center">
            <xsl:text>Strana </fo:page-number/>
          </fo:block>
        </fo:static-content>
        <!-- Text dokumentu -->
        <fo:flow flow-name="xsl-region-body">
          <!-- Zpracování elementu TEXT -->
          <xsl:apply-templates/>
          <!-- Načtení souborů -->
          <xsl:apply-templates select="document(//file)
/node() "/>
        </fo:flow>
      </fo:page-sequence>
    </fo:root>
  </xsl:template>

  <xsl:template match="odkaz">
  </xsl:template>

  <!-- Transformace elementu LOGO1 -->
  <xsl:template match="logol">
    <fo:block text-align="center">
      <fo:external-graphic src="{@url}"/>
    </fo:block>
  </xsl:template>
</xsl:stylesheet>
```



```

        </fo:block><fo:block></fo:block>
</xsl:template>

<!--Transformace elementu NazevCZ1 -->
<xsl:template match="nazevcz1">
    <fo:block text-align="center">
        <xsl:text>Provozně ekonomická fakulta</xsl:text>
    </fo:block><fo:block></fo:block>
    <fo:block font-size="24pt" font-weight="bold"
        text-align="center">
        <xsl:apply-templates/>
    </fo:block>
</xsl:template>

<!--Transformace elementu NazevEN1 -->
<xsl:template match="nazeven1">
    <fo:block></fo:block>
    <fo:block text-align="center">
        <xsl:apply-templates/>
    </fo:block><fo:block></fo:block>
    <fo:block text-align="center"><xsl:value-of select=
"/pracel/@typ"/></fo:block>
</xsl:template>

<!--Transformace elementu Autor1 -->
<xsl:template match="autor1">
    <fo:block></fo:block>
    <fo:block text-align="end" space-before="6pt">
        <xsl:text>Autor: </xsl:text><xsl:apply-templates/></fo:block>
</xsl:template>

<!--Transformace elementu Vedouci1 -->
<xsl:template match="vedouci1">
    <fo:block text-align="end" space-before="6pt">
        <xsl:text>Vedoucí: </xsl:text><xsl:apply-templates/>
    </fo:block>
</xsl:template>

<!-- Transformace prvnioho vyskytu elementu NADPIS11 -->
<xsl:template match="nadpis11[1]">
    <!-- Vlozi zalomeni konec stranky -->
    <fo:block break-before="page"/>
    <fo:block font-weight="bold"><xsl:text>Klíčová
slova:</xsl:text></fo:block><fo:block><xsl:value-of select=
"/pracel/@key"/></fo:block>
    <fo:block font-weight="bold"><xsl:text>Key
words:</xsl:text></fo:block><fo:block><xsl:value-of select=
"/pracel/@keyen"/></fo:block>
    <fo:block></fo:block>
    <fo:block text-align="left" font-size="20" font-weight="bold">
        <xsl:text>Osnova: </xsl:text><fo:block></fo:block>
    </fo:block>
    <fo:block text-align="left" font-size="20" font-weight="bold">
        <xsl:apply-templates/>
    </fo:block>
</xsl:template>

<!-- Transformace elementu NADPIS11 -->
<xsl:template match="nadpis11">

```

```

        <fo:block text-align="left" font-size="20" font-weight="bold">
            <xsl:apply-templates/>
        </fo:block>
    </xsl:template>

    <!-- Transformace elementu NADPIS21 -->
    <xsl:template match="nadpis21">
        <fo:block text-align="left" font-size="18" font-weight="bold">
            <odsazeni><odsazeni><xsl:apply-
templates/></odsazeni></odsazeni>
        </fo:block>
    </xsl:template>

    <!-- Transformace elementu NADPIS31 -->
    <xsl:template match="nadpis31">
        <fo:block text-align="left" font-size="16" font-weight="bold">
            <xsl:apply-templates/>
        </fo:block>
    </xsl:template>

    <!-- Transformace elementu PRACE1 -->
    <xsl:template match="prace1">
        <xsl:apply-templates/>
        <fo:block break-before="page"/>
    </xsl:template>

    <!-- Transformace elementu UVOD -->
    <xsl:template match="uvod">
        <fo:block><xsl:apply-templates/></fo:block>
    </xsl:template>

    <!-- Transformace elementu KAPITOLA -->
    <xsl:template match="kapitola">
        <fo:block break-before="page"/>
        <fo:block>
            <xsl:apply-templates/>
        </fo:block>
    </xsl:template>

    <!-- Transformace elementu ZAVER -->
    <xsl:template match="zaver">
        <fo:block break-before="page"/>
        <fo:block>
            <xsl:apply-templates/>
        </fo:block>
    </xsl:template>

    <!-- Transformace elementu NADPIS1 -->
    <xsl:template match="nadpis1">
        <fo:block text-align="left" font-size="20" font-weight="bold">
            <xsl:apply-templates/>
        </fo:block>
    </xsl:template>

    <!-- Transformace elementu NADPIS2 -->
    <xsl:template match="nadpis2">
        <fo:block text-align="left" font-size="18" font-weight="bold">
            <xsl:apply-templates/>
        </fo:block>
    </xsl:template>

```

```

</xsl:template>

<!-- Transformace elementu NADPIS3 -->
<xsl:template match="nadpis3">
  <fo:block text-align="left" font-size="16" font-weight="bold">
    <xsl:apply-templates/>
  </fo:block>
</xsl:template>

<!--Transformace elementu P -->
<xsl:template match="p">
  <!-- Velikost pisma 12, písmo Times New Roman, odsazení začátku
30pt, zarovnání do bloku, řádkování 150% -->
  <fo:block font-size="12" font="Times New Roman" text-
indent="30pt" text-align="justify" line-height="150%">
    <xsl:apply-templates/>
  </fo:block>
</xsl:template>


<!-- Transformace elementu IMG -->
<xsl:template match="img">
  <fo:block></fo:block>
  <fo:block text-align="{@zarovnaní}">
    <fo:external-graphic src="{@url}"/>
  </fo:block><fo:block text-align="{@zarovnaní}"><xsl:value-of
select= "@alt"/></fo:block><fo:block></fo:block>
</xsl:template>

<xsl:template match="nazevcz">
</xsl:template>
<xsl:template match="nazeven">
</xsl:template>
<xsl:template match="autor">
</xsl:template>
<xsl:template match="file">
</xsl:template>
<xsl:template match="vedouci">
</xsl:template>
</xsl:stylesheet>

```

Výsledkem transformace je soubor dp1.fo, který je již připraven pro tisk. K otevírání souborů formátů .fo poslouží například software Software602 Form Filler, v kterém jsem výstup kontroloval. Jelikož rozdílné procesory mohou kód interpretovat různým způsobem (jako například webová stránka zobrazená v různých prohlížečích), převedl jsem výstupní soubor i do formátu pdf (dp1.pdf). Ke konverzi jsem použil také výše zmíněný Software602 Form Filler.

Na obrázku č. 6 je vidět náhled prvních čtyř stránek v programu Adobe Reader.

 <p>ČESKÁ ZEMĚDĚLSKÁ UNIVERZITA V PRAZE</p> <p>Provozní ekonomická fakulta</p> <p>XML a OpenDocument Format standards a jejich použití v praxi</p> <p>XML and OpenDocument Format standards and their use in practice</p> <p>DP</p> <p>Autor: Jan Dostál Vedoucí: Ing. Česzník Halbich CSc.</p> <p>Strana 1</p>	<p>Klíčová slova: XML DTD XSLT XPath XLink XPointer Filter602 602Designer OpenDocument</p> <p>Key words: XML DTD XSLT XPath XLink XPointer Filter602 602Designer OpenDocument</p> <p>Osnova:</p> <ol style="list-style-type: none">1. Úvod2. Cíl práce a metodika3. XML4. OpenDocument5. Použití v praxi<ol style="list-style-type: none">5.1 XML5.2 602 Designer5.3 OpenDocument6. Závěr7. Seznam použitých zdrojů8. Přílohy <p>Strana 2</p>
<p>1. Úvod Zde se bude nacházet úvod</p> <p>Strana 3</p>	<p>2. Cíl práce a metodika</p> <p>Strana 4</p>

Obrázek č. 6- DP1.fo

5.1.4 Zhodnocení XML

Při zhodnocení začnu u DTD. Ačkoliv je výše uvedené DTD poměrně jednoduché, při tvorbě následných schémat jsem několikrát narazil na problém a musel některé elementy, případně atributy upravovat.

Pokud by soubor s DTD byl již rozšířen mezi studenty odevzdávající práce v XML formátu, jeho následná aktualizace by byla o poznání složitější, proto je vhodné ho pořádně vyzkoušet na praktických příkladech.

Jednou ze základních myšlenek XML je oddělení informačního obsahu dokumentu od definice grafického vzhledu. Jak bude dokument vypadat se definuje pomocí stylových jazyků, například XSLT. V mém projektu se XSLT ukázalo jako velmi silné. U této technologie je důležité hlavně využívání jazyka XPath, jenž dokáže vybrat část dokumentu. Na vybranou část lze poté aplikovat bez větších problémů styl či transformaci. Při transformování do jiných formátů než XML je, dle mého názoru, problémem znalost struktury a zákonitostí výstupního souboru, ne správná syntaxe XSLT.

Nakonec FO, formátovací objekty. Velkou výhodou FO vidím v podobnosti s rozšířeným jazykem CSS, což minimálně ze začátku usnadní tvorbu. Ani tato, ani jiná výhoda však nepřebije v současnosti hlavní nevýhodu- chybějící podporu. S tím se však před lety potýkala i XSLT schémata, která dnes dokáže interpretovat každý rozšířený webový prohlížeč. Jazyk není pro ruční psaní složitý, avšak obsahuje již velké množství značek. Proto je pro tvorbu vhodné místo textového editor využít nějaký specializovaný editor.

XML je velmi mocný jazyk. Jeho schopnosti jsou však přímo závislé na dalších používaných technologiích a tak nelze nalézt jeho podstatné chyby, či vylepšení.

5.2 602XML Designer

V této části diplomové práce převedu do elektronické podoby tištěný formulář. Původně jsem chtěl vytvořit formulář pro vyplňování daňového přiznání, ovšem zjistil jsem, že takový formulář firma Software602 svým zákazníkům již nabízí. Proto tedy převedu formulář společnosti PRE Žádost o změnu / uzavření / ukončení smlouvy (viz. příloha č. 2). K vytvoření elektronické verze formuláře využiji software 602XML Designer, jehož licenci jsem získal právě za účelem tvorby této diplomové práce.

Formulář je využíván pro šest různých činností (převod, ukončení odběru, změna platebního způsobu, elektronická fakturace, změna ostatních údajů, změna zásilací adresy). Podle jednotlivých důvodů žádosti se vždy vyplňuje pouze jeho část. Proto je formulář rozdělen na osm částí.

Na tyto části jsem rozdělil i elektronickou verzi. V následujícím textu budu popisovat jen tvorbu oddílu „identifikace stávajícího odběratele a odběrného místa“.

5.2.1 Tvorba formuláře

Prvním krokem při tvorbě formuláře byla tvorba jeho vzhledu. Nejednalo se o nic složitého, jak je vidět na obrázku č. 7, k přesnému umístění jednotlivých prvků dobře posloužily tabulky.

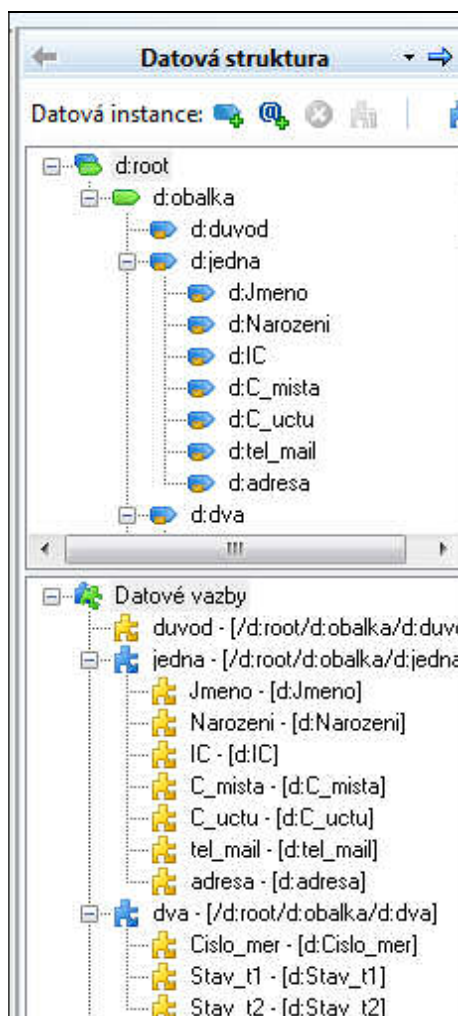
The image shows a screenshot of the 602XML Designer software interface. The main window displays a form layout for the section "Identifikace stávajícího odběratele a odběrného místa". The form is structured as follows:

- A header row with the title "Identifikace stávajícího odběratele a odběrného místa".
- A row for "jméno/ název / firma".
- A row for "Datum narození" and "IČ".
- A row for "Tel. / email".
- A row for "Číslo odběrného místa" (with a note "naleznete na fakturě") and "Číslo zákaznického účtu" (with a note "naleznete na fakturě").
- A row for "Adresa odběrného místa".

The interface also shows XML tags on the left side, indicating the structure of the form elements: fo:block, fo:table, and /fo:table.

Obrázek č. 7-Vzhled

Další částí realizace bylo vytvoření datové struktury. Ve vrchní části tabulky na obrázku č. 8 probíhá vytváření elementů, ve spodní části je k nim poté přiřazována datová vazba.



Obrázek č. 8- Datová struktura

Po vytvoření vhledu schránky a její datové struktury následuje spojení těchto dvou částí- konvertace jednotlivých částí tabulky na formulářové prvky.

602XML Designer nabízí možnost měnit jednotlivé elementy na textové pole, číselné pole, zaškrtačací pole, přepínač, rozbalovací seznam, pole pro vkládání času, pole pro vkládání data, buňkové pole, buňkové číselné pole nebo na tlačítko.

V popisovaném bloku jsem použil tyto konvertace:

Jméno- textové pole; bez dalších omezení

Datum narození- textové pole; zobrazený text DD.MM.RRRR; maximální délka 10 znaků; povolené znaky číslice a tečky. Jako typ nebylo vybráno pole pro vkládání data, jelikož toto pole při vkládání zobrazí „kalendář“, v kterém by uživatel těžko listoval několik let zpátky.

IČ- textové pole; povolené znaky číslice

Číslo odb. místa- buňkové číselné pole; deset polí; v prvních třech buňkách hodnoty 8-1-1

Číslo zákaz účtu- buňkové číselné pole; 8 polí

Adresa odběrného místa- textové pole; bez dalších omezení

Identifikace stávajícího odběratele a odběrného místa		
Jméno/ název / firma		
Datum narození		
Tel. / email		
Číslo odběrného místa	(naleznete na faktuře)	8 1 1
Číslo zákaznického účtu	(naleznete na faktuře)	
Adresa odběrného místa		

Obrázek č. 9- Formulářové prvky

Elementy, které jsem změnil na formulářové prvky jsou nyní označeny žlutou barvou, jak je vidět na obrázku č. 9.

V programu lze zobrazit XPath jednotlivých prvků. Pro zajímavost nyní uvádím přesnou adresu části tabulky, v které je vepsáno DD.MM.RRRR.

```
/fo:root/fo:page-sequence[1]/fo:flow[1]/fo:block[1]/fo:block[2]/fo:multi-switch[1]/fo:multi-case[2]/fo:block[1]/fo:table[3]/fo:table-body[1]/fo:table-row[1]/fo:table-cell[2]
```

Uvedený postup jsem aplikoval i na další části formuláře, zde stojí za zmínku hlavně použití rozbalovacího seznamu a s ním spojenou tvorbu číselníků.

Dalším krokem bylo vytvoření titulní stránky, na které uživatel volí, za jakým účelem formulář vyplňuje. Titulní stránka je vidět na obrázku č. 10.

PRE Žádost o změnu / uzavření / ukončení smlouvy

Důvod žádosti

Změna odběratele (převod) ¹⁾

Ukončení odběru ²⁾

Změna platebního způsobu

Elektronická fakturace ⁷⁾

Změna ostatních údajů

Změna zasilací adresy

Vysvětlivky:

1) Převodem se rozumí změna odběratele v odběrném místě specifikovaném v této žádosti bez přerušení dodávky elektřiny a bez změn charakteru odběru. Stávající odběratel na základě této žádosti o ukončení dodávky elektřiny končí odběr elektřiny v odběrném místě. Místo něho bude v odběrném místě specifikovaném v této žádosti odbírat elektřinu od dodavatele Pražské energetiky a.s., nový odběratel na základě této žádosti o uzavření smlouvy o sdružených službách dodávky elektřiny. V případě, že nebude zahájena dodávka elektřiny novému odběrateli, dojde na základě této žádosti pouze k ukončení dodávky elektřiny stávajícímu odběrateli do 30 dnů od doručení žádosti.

2) K ukončení smlouvy na dodávku elektřiny z důvodu taktického ukončení odběru zákazníkem dojde provedením konečného odečtu stavu elektroměru do 30 dnů od doručení této žádosti. Dodavatel následně vystaví zákazníkovi konečnou fakturu. Není-li elektroměr přístupný, nemůže být odběr ukončen, a je proto v zájmu zákazníka dohodnout termín návštěvy a zajistit přístupnost elektroměru. I v případě, že je elektroměr přístupný, doporučujeme zákazníkům, aby byli konečnému odečtu přítomni. Vzájemné závazky budou vyrovnány v režimu stávající platební metody, pokud nebude dohodnuto jinak.

7) V případě požadavku na vystavování faktur v elektronické podobě není zaslána faktura v písemné podobě.

Zákaznická linka PRE	Zákaznické centrum PRE	Centrum energetického poradenství PRE
Tel.: 267 055 555	Praha 1, Jungmannova 31 (palác Adria)	Jungmannova 28 (palác Adria), P1
Fax: 267 055 505	Praha 4, Vladimírova 18	Provozní doba: Po-Pá 10.00-18.00
E-mail: pre@pre.cz	Praha 9, Ocelářská 5s	Tel.: 267 055 555
Provozní doba: Po-pá 7.00-19.00	Provozní doba: Po-Čt 9.00-18.00	E-mail: poradce@pre.cz
Záznamník: 19.00-7.00	Pá 9.00-12.00	www.energetickyporadce.cz

Zákaznické e-centrum PRE
www.pre.cz/e-centrum
www.pre.cz

Obrázek č. 10- Titulní strana

Výběr je realizován jako takzvaný oddíl s přepínáním, čili pomocí funkce fo:multi-switch, kde jednotlivé části jsou uvnitř objektů fo:multi-case. Šestice multi-case tak koresponduje s šesticí důvodů žádosti.

Předposledním krokem tvorby bylo nakopírování jednotlivých částí formuláře do bloků multi-case, v kterých jsou vyžadovány.

Závěrečnou částí realizace převodu tištěného formuláře na formulář elektronický byla zevrubná zkouška funkčnosti. Jednalo se hlavně o zkoušku toho, zda do některého prvku nejde vložit zakázaný znak.

Výsledný soubor pre.fo obsahuje velmi mnoho formátovacích značek a proto by jeho zkopírování zabralo několik desítek stránek. Vložím alespoň malou část, na které je dobře vidět, že se jedná o formát XSL:FO, který by bylo možné vytvořit v pouhém textovém editoru. K této části jsem přidal komentáře

Část souboru pre.fo:

```
<!-- Začátek části, která se zobrazí po výběru příslušného case, zde
Zmena_odb -->
<fo:multi-case case-name="Zmena_odb" form602ct="cas">

  <!-- Společné vlastnosti pro celý blok; velikost písma 9, tučné -->
  <fo:block font-size=" 9pt" font-weight="bold">

    <!-- Vlastnosti pro prvky v tabulce -->
    <fo:table table-layout="fixed" font-size="12pt" margin-
bottom="2pt" font-weight="bold">

      <!-- Velikost jednotlivých sloupců -->
      <fo:table-column column-width="11cm"/><fo:table-column
column-width="1mm"/>

      <!-- Tělo tabulky -->
      <fo:table-body>

        <!-- Řádek tabulky -->
        <fo:table-row>
          <fo:table-cell background-color="darkblue"
color="white">
            Identifikace stã~vajã-cã-ho odbã>ratele a
odbã>rnã©ho mã-sta
          </fo:table-cell>
          <fo:table-cell background-color="darkblue"
color="white"/>
```

```

        </fo:table-row>
    </fo:table-body>
</fo:table>
<fo:table table-layout="fixed">
    <fo:table-column column-width="3.9cm"/><fo:table-column/>
    <fo:table-body>
        <fo:table-row>
            <fo:table-cell>Jméno / název / firma</fo:table-cell>
            <fo:table-cell form602="Jmeno"
                form602bind="Jmeno" form602ct="txt"/>
        </fo:table-row>
    </fo:table-body>
</fo:table>
<fo:table table-layout="fixed" margin-bottom="2pt" margin-
top="2pt">
    <fo:table-column column-width="3.9cm"/><fo:table-column/>
    <fo:table-column column-width="1cm"/><fo:table-column/>
    <fo:table-body>
        <fo:table-row>
            <fo:table-cell text-align="start" display-
align="center">Datum narození</fo:table-cell>

        <!-- Pole tabulky je zároveň formulářový prvek
Narozeni; předvyplněný text DD.MM.RRRR; maximální délka 10 znaků; možno
číslice nebo tečky -->
            <fo:table-cell form602="Narozeni"
form602text="DD.MM.RRRR" form602maxchars="10"
form602chars="decimal_digits|punctuation" form602bind="Narozeni"
form602ct="txt" text-align="center"/>
            <fo:table-cell text-align="center">IČ</fo:table-
cell>

        <!-- Pole tabulky je zároveň formulářový prvek IC;
povolené znaky číslice a mezery -->
            <fo:table-cell form602="IC"
form602chars="decimal_digits|space_characters" form602bind="IC"
form602ct="txt" text-align="center"/>
        </fo:table-row>
    </fo:table-body>
</fo:table>
<fo:table table-layout="fixed">
    <fo:table-column column-width="3.9cm"/><fo:table-column/>
    <fo:table-body>
        <fo:table-row><fo:table-cell display-align="center">Tel.
/ email</fo:table-cell>
            <fo:table-cell form602="tel_mail"
form602bind="tel_mail" form602ct="txt" text-align="center"/>
        </fo:table-row>
    </fo:table-body>
</fo:table>

```

U výsledného formuláře je nakonec možné nastavit, co se s ním bude dít po odeslání, například digitálně podepsat a uložit jako PDF, vytisknout, či odeslat na mail.

5.2.2 Výsledné formuláře

Výsledkem uvedené práce není jeden formulář, ale dle volby na titulní stránce formulářů šest. Na obrázku č. 11 je vidět vzhled hlavní části vyplněného formuláře Změna odběratele.

IPRE		Žádost o změnu / uzavření / ukončení smlouvy	
Identifikace stávajícího odběratele a odběrného místa			
Jméno/ název / firma	Jan Dostál		
Datum narození	24.12.1900	IČ	27
Tel. / email	xdosj301@pef.czu.cz		
Číslo odběrného místa <small>(naleznuté na faktuře)</small>	8 1 1	1 2 3 4 5 6 7	
Číslo zákaznického účtu <small>(naleznuté na faktuře)</small>	8 9 1 2 3 4 5 6		
Adresa odběrného místa	Podzemní 7, Praha 6		
Údaje o měřicím zařízení			
Číslo elektroměru	007	Stav elektroměru uvedený odběratelem ³⁾	T1 24 T2 33
Adresa pro zaslání faktur a korespondence		<small>(Adresa bude použita pro zaslání faktury)</small>	
Jméno / název / firma	Jan Dostál	Datum narození	24.12.1900
Ulice	Podzemní	Č. popisné	12
Obvod	Kbely	Obec	Praha
Číslo bankovního účtu k vyrovnání závazků		PSČ	16178
			21110540044/0100
<p>Stávající odběratel žádá společnost PREdistribuce, a.s., aby ke dni ukončení odběru elektřiny v odběrném místě dle této žádosti byla ukončena účinnost smlouvy o připojení pro toto odběrné místo. Pražská energetika, a.s. se zavazuje předat uvedenou žádost společnosti PREdistribuce, a.s. IČ 27376516. Prohlašuji, že mnou uvedené údaje jsou pravdivé a souhlasím s nimi, což stvrzuji vlastnoručním podpisem.</p>			
Podpis stávajícího odběratele: v Praze dne 10.2.2011			
Nový odběratel			
Jméno a příjmení / název / firma	Zelenina Novák		
Trvalé bydliště / sídlo / Ulice	Masná	Č. orient. ⁴⁾	47
/ místo podnikání	Obvod	Obec	Praha
Datum narození	01.01.1950	IČ	
Jednající / zástupce ⁵⁾	Jan Novák		
Kontakty: <small>(povinný údaj)</small>	Telefon (mobil)	606123456	Fax
			e-mail novak@seznam.cz
Požadovaná distribuční sazba ⁶⁾		účel odběru	odběr v kategorii D
Elektronická fakturace: ⁷⁾	e-mail	novak@seznam.cz	
Zasílací adresa	trvalé bydliště / sídlo / místo podnikání		
Platební způsob			
Bezhotovostně dle vlastního výběru (prostřednictvím banky, pošty atd.). Požadujete-li některý z dále uvedených způsobů, vyberte ho a vyplňte příslušné údaje.			
Příkaz k úhradě			
Frekvence záloh ⁸⁾ půlročně			

Obrázek č. 11- Změna odběratele

Na obrázku č. 12 je vyplněný formulář Změna zasílací adresy.

PRE		Žádost o změnu / uzavření / ukončení smlouvy			
Identifikace stávajícího odběratele a odběrného místa					
Jméno/ název / firma	Jan Dostál				
Datum narození	01.01.1911	IČ	27		
Tel. / email					
Číslo odběrného místa (naleznete na faktuře)	8 1 1	1 1 1 1 1 1 1 1			
Číslo zákaznického účtu (naleznete na faktuře)		1 1 1 1 1 1 1 1			
Adresa odběrného místa	Přítopení 8, Praha				
Adresa pro zasílání faktur a korespondence					
Jméno / název / firma	Jan Dostál	Datum narození	01.01.1911		
Ulice	Kotvá	Č. popisné	47	Č.orientační	45
Obvod	2	Obec	Praha	PSČ	12000
Číslo bankovního účtu k vyrovnání závazků					454874421/0300
Prohlašuji, že mnou uvedené údaje jsou pravdivé a souhlasím s nimi, což stvrzuji vlastnoručním podpisem.					
Podpis stávajícího odběratele: v Praze dne 2.3.2011					
Převzal zástupce Pražské energetiky a.s..... dne					

Obrázek č. 12- Změna zasílací adresy

Výsledný formulář byl předán zaměstnancům firmy PRE, kde v současnosti probíhá posouzení, zda ho nevyužít v praxi.

5.2.2 Zhodnocení 602XML Designer

Program 602XML Designer nabízí uživatelům intuitivní ovládaní, díky čemuž je práce s ním velmi příjemná. Tvořit formuláře lze i bez obsáhlých znalostí XML a přidružených technologií.

Elektronizace dokumentů musí splňovat tyto požadavky:

- výsledný dokument má stejnou právní účinnost, jako dokument původní
- při vytváření dokumentu dochází oproti tištěnému dokumentu k úspoře práce či času
- zadaná data jsou připravena pro další zpracování

Tvorba formulářů pomocí 602XML Designer tyto požadavky splňuje. Dokument lze opatřit elektronickým certifikátem či elektronickým časovým razítkem, čímž je zaručena právní účinnost. Online vyplňování jistě ušetří mnoho času, mnoho údajů si formuláře mohou dopočítat automaticky, čímž se ušetří práce. Automatická kontrola vstupních dat poté zaručí převoditelnost dokumentu do dalších formátů.

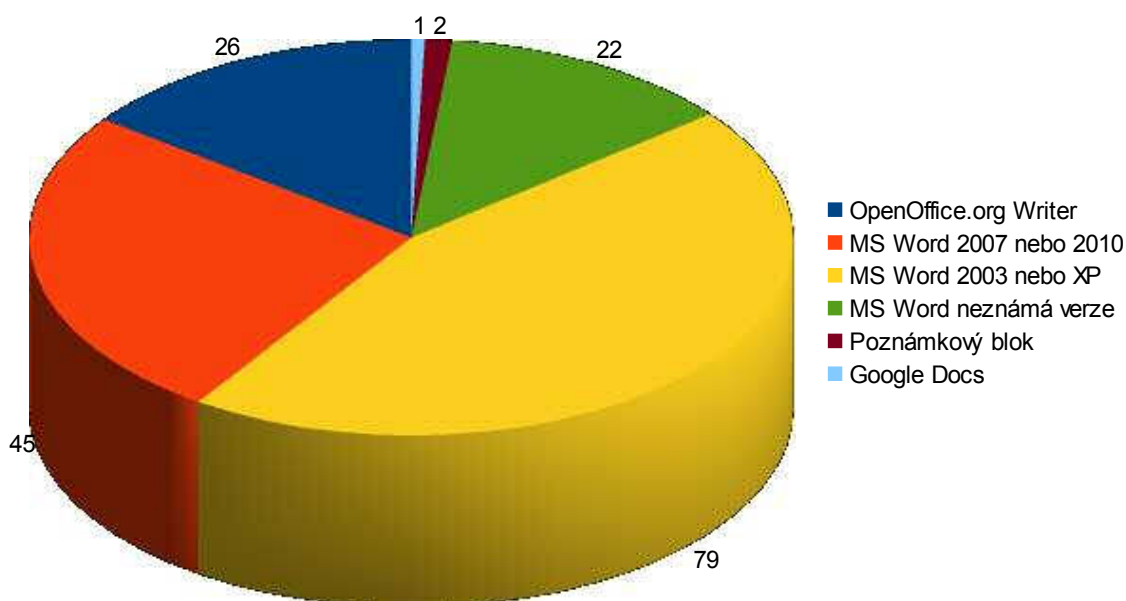
Největším problémem 602XML Designeru, potažmo XSL:FO, je chybějící podpora. Běžné programové vybavení osobních počítačů k vyplnění formuláře nestačí. K správnému používání je třeba nainstalovat Software602 Form Filler, čímž je na druhou stranu omezena nesprávná interpretace kódu a zaručena správná funkčnost.

5.3 Porovnání kancelářských formátů

V této kapitole porovnám ODF používaný pro textové dokumenty, soubor typu .odt, s používanou konkurencí.

5.3.1 Výběr konkurence

Pomocí aplikace Google Docs jsem vytvořil dotazník (viz příloha č. 1), jehož první otázka se týkala používaného textového editoru. Po vyřazení nesmyslných odpovědí mi zůstal vzorek 179 použitelných. Na obrázku č. 12 je vidět, jaký textový editor uživatelé používají.



Obrázek č. 13- Používané editory

Z grafu na obrázku č. 5.3.1.1 je jasně patrná převaha různých verzí textového editoru Microsoft Office Word. Mezi editory, které jako základní formát využívají ODF pak jasně dominuje OpenOffice.org Writer.

5.3.2 Porovnání editorů a formátů

Na základě grafu z obrázku č. 12 jsem do srovnání zahrnul textové editory MS Word XP, MS Word 2003, MS Word 2007, MS Word 2010 a OpenOffice.org Writer.

Jelikož další část dotazníkového průzkumu neukázala nějakou důležitou funkci, díky které by uživatelé některý produkt preferovali, pro testování jsem vytvořil dokument obsahující běžně používané funkce a objekty. Testování jsem provedl pomocí jedné ze základních empirických metod- experimentování.

Porovnání bude probíhat následujícím způsobem:

- Nejprve v různých editorech vytvořím shodný dokument. Přitom budu sledovat hlavně, jestli editor potřebně funkce obsahuje.
- Poté budu vytvořené soubory otvírat v různých editorech. Pokud se soubor podaří otevřít, budu kontrolovat, zda se jednotlivé prvky zobrazily správně.
- Na základě získaných poznatků poté vyvodím závěry.

Vytvořený soubor obsahuje tyto části, které budu kontrolovat:

- napsaný text
- vložený obrázek
- vložený obrázek a jeho obtékání textem
- vložený graf
- vložená tabulka
- formátování v tabulce
- vytvořený obsah, fungující jako odkazy
- nastavení stylů
- nastavení velikosti stránky
- zápatí obsahující čísla stránek
- dva oddíly, přičemž stránky jsou číslovány až od druhého oddílu
- vložení zalomení „konec stránky“

Vyjmenované prvky byly schopny do souboru vložit všechny testované editory.

V tabulce č. 4 jsou křížkem označeny programy, v nichž lze soubor z prvního sloupce otevřít. V prvním sloupci jsou tučným písmem zvýrazněny formáty, které editor používá standardně, normálním písmem pak testované formáty, do kterých je program schopen ukládat.

	OpenOffice.org Writer	Word XP	Word 2003	Word 2007_sp2	Word 2010
OpenOffice.odt	X			X	X
OpenOffice.doc	X	X	X	X	X
Word_XP.doc	X	X	X	X	X
Word_2003.doc	X	X	X	X	X
Word_2007.doc	X	X	X	X	X
Word_2007.docx				X	X
Word_2007.odt	X			X	X
Word_2010.doc	X	X	X	X	X
Word_2010.docx				X	X
Word_2010.odt	X			X	X

Tabulka č. 4- Možnosti otevírání

Z tabulky je patrné, že z testovaných formátů lze ve všech zkoušených programech otevřít pouze starší formát Microsoftu .doc. Podpora ODF byla navíc do Microsoft Office 2007 implementována až s opravným balíčkem SP2. Podle obchodních sdělení má být po instalaci příslušného modulu Word 2003 schopen otevřít dokumenty .docx. Při testování se však pouze změnila chybová hláška z „neznámého formátu“ na „chyba při otevírání souboru“. Nejuniverzálnější se tedy zdají editory Microsoft Office Word 2007 a 2010, které jsou schopny otevřít všechny testované soubory.

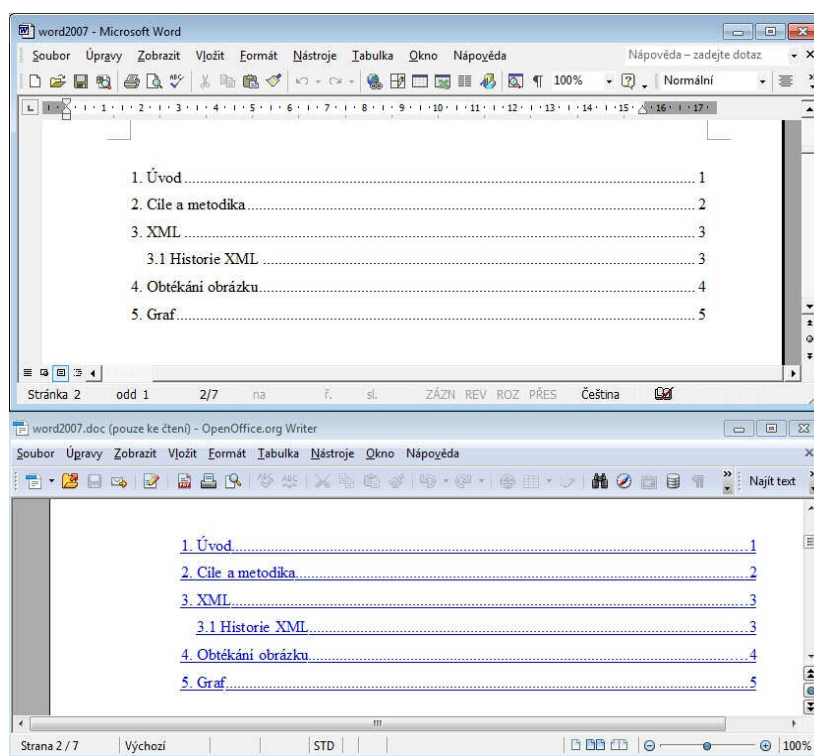
Schopnost otevřít soubor však není jedinou důležitou vlastností v oblasti kompatibility. Důležité je také to, zda se zobrazí všechny prvky a zda jsou tyto prvky správně formátovány.

Pokud se podařilo soubor otevřít, editory neměly s většinou prvků problémy. Správně se zobrazovaly obrázky, včetně jejich obtékání, text, tabulka, formátování obsahu tabulky, oddíly. Správně byla nastavena i velikost stránky a formátování používaných stylů. I zalomení konec stránky zafungovalo vždy dle požadavků

Problém byl se zápatím dokumentu OpenOffice.org_Writer.odt vytvořeného pomocí OpenOffice.org Writeru. Word ve verzi 2007, ani 2010 toto zápatí nezobrazil. Zajímavé je, že .odt soubory vytvořené Wordem 2007 i 2010 zápatí s číslem stránek taktéž obsahovaly a OpenOffice.org Writer toto zápatí zobrazil správně.

Další problém nastal ve Wordu 2007 a 2010 při ukládání do staršího formátu .doc. Graf nebyl uložen správně. Na tuto možnost jsem byl při ukládání upozorněn. Špatně uložený graf pak logicky nešlo správně otevřít.

Největší nesrovnalosti se ukázaly při tvorbě obsahu práce. Při použití Writeru na otevírání souborů vytvořených jakoukoliv verzí MS Word, je obsah na rozdíl od zdroje vypsán modrou barvou a jsou vykresleny vodící čáry. Na obrázku č. 14 je vidět popisovaná část souboru word2007.doc. Nahoře podoba v MS Word XP, dole pak ve Writeru.



Obrázek č. 14- Zobrazený obsah

Obdobně špatně, navíc se šedivým pozadím, byla obsahová část dokumentu OpenOffice.org_Writer.odt zobrazena ve Wordu 2007 i 2010.

Na závěr ještě jedna zajímavá vlastnost porovnávaných formátů. Ačkoliv jsou dokumenty po obsahové stránce shodné, formát .doc zabírá zhruba dvakrát tolik místa, než formáty .docx a .odt.

5.3.3 Dotazník

Dotazníkový průzkum přinesl i jiné zajímavé informace, než jen používaný editor.

U uživatelů MS Office jsem se zajímal, zda používají legální verzi. Podle odpovědí legální verzi používá 91 ze 146 uživatelů MS Office, 41 jich používá verzi nelegální, zbylých 14 uživatelů neví.

Zajímavé jsou taktéž důvody používání Wordu. Ačkoliv se o žádný standard nejedná, pro 101 ze 146 respondentů je to hlavní důvod, proč ho používají. Dalších 24 nemá na výběr, pro 20 uživatelů jsou hlavním důvodem lepší funkce. Ačkoliv k tomu byl v dotazníku prostor, bohužel tyto funkce nikdo nespécifikoval. Pro jednoho respondenta je poté důvodem k používání Wordu jeho pohodlnost.

I když ODF standardem je, žádný z 26 uživatelů to neuvedl jako hlavní důvod pro používání OpenOffice.org Writeru. 22 z 26 (což je 91%) uživatelů Writer využívá, protože je zadarmo. Jeden uživatel z důvodu lepší a jednodušší úpravy grafů.

5.3.4 Zhodnocení ODF

Podle literární rešerše by se mohlo zdát, že OpenDocument Format je dlouho potřebný všelék, který okamžitě odstraní problémy uzavřených formátů. Z dlouhodobého hlediska bude jeho používání jistě velice prospěšné, ba dokonce potřebné.

Avšak ačkoliv k tomu po technické stránce neexistuje důvod, v současnosti hrají prim uzavřené formáty firmy Microsoft. To nezměnil ani přechod na nový formát .docx.

Problémy s kompatibilitou mezi verzemi MS Office, používajícími starý a nový formát, však naznačují, jaká časovaná bomba se v používání uzavřených formátů skrývá.

Bez produktů Microsoft Office se většina kanceláří v současnosti zřejmě neobejde. Dle mého názoru by však měl být otevřený formát ODF podporován například ve školství. S lepšími vyhlídkami do budoucna a nižší cenou totiž dokáže nabídnout stejné funkce, jako formát MS Office.

6. Závěr

Tři části měla praktická část této diplomové práce, s tím budou korespondovat i tři části závěru

Při studování odborné literatury a hlavně při tvorbě praktické části diplomové práce jsem zjistil, že XML je velmi mocný jazyk, určený hlavně pro výměnu dat mezi aplikacemi a pro publikování dokumentů. Jeho schopnosti jsou však přímo závislé na dalších používaných technologiích a tak nelze nalézt jeho podstatné chyby, či vylepšení.

Jednou ze základních myšlenek XML je oddělení informačního obsahu dokumentu od definice grafického vzhledu. Jak bude dokument vypadat se definuje pomocí stylových jazyků, například XSLT. V projektu se ukázala schémata XSLT jako velmi silná. U této technologie je důležité hlavně využívání jazyka XPath, jenž dokáže vybrat část dokumentu. Na vybranou část lze poté aplikovat styl či transformaci bez větších problémů. Při transformování do jiných formátů než XML je, dle mého názoru, problémem spíše znalost struktury a zákonitostí výstupního souboru, ne správná syntaxe XSLT.

Nakonec FO, formátovací objekty. Velkou výhodu FO vidím v podobnosti s rozšířeným jazykem CSS, což minimálně ze začátku usnadní tvorbu. Ani tato, ani jiná výhoda však nepřebije v současnosti hlavní nevýhodu- chybějící podporu. S tím se však před lety potýkala i XSLT, která dnes dokáže interpretovat každý rozšířený webový prohlížeč. Jazyk není pro ruční psaní složitý, avšak obsahuje již velké množství značek. Proto je pro tvorbu vhodné místo textového editor využít nějaký specializovaný editor.

Druhá část se věnovala vytvářením formulářů pomocí komerčního programu 602XML Designer. Byl elektronizován formulář, u kterého nyní zaměstnanci firmy Pražská energetika, a.s. posuzují přínos a případné použití.

Nástroj 602XML Designer nabízí uživatelům intuitivní ovládaní, díky čemuž je práce s ním velmi příjemná. Tvořit formuláře lze i bez obsáhlých znalostí XML a přidružených technologií.

Tvorba formulářů pomocí 602XML Designer splňuje požadavky elektronizace dokumentů. Dokument lze opatřit elektronickým certifikátem či elektronickým časovým razítkem, čímž je zaručena právní účinnost. Online vyplňování jistě ušetří mnoho času,

mnoho údajů si formuláře mohou dopočítat automaticky, čímž se ušetří práce. Automatická kontrola vstupních dat poté zaručí převoditelnost dokumentu do dalších formátů.

Největším problémem 602XML Designeru, potažmo XSL:FO, je chybějící podpora. Běžné programové vybavení osobních počítačů k vyplnění formuláře nestačí. K správnému používání je třeba nainstalovat Software602 Form Filler, čímž je na druhou stranu omezena nesprávná interpretace kódu a zaručena správná funkčnost.

Poslední, třetí část se vztahovala k formátu ODF. Podle literární rešerše by se mohlo zdát, že OpenDocument Format je dlouho potřebný všelék, který okamžitě odstraní problémy uzavřených formátů. Z dlouhodobého hlediska bude jeho používání jistě velice prospěšné, ba dokonce potřebné.

Avšak ačkoliv k tomu po technické stránce neexistuje důvod, v současnosti hrají prim uzavřené formáty firmy Microsoft. To nezměnil ani přechod na nový formát .docx. Problémy s kompatibilitou mezi verzemi MS Office, používajícími starý a nový formát, však naznačují, jaká časovaná bomba se v používání uzavřených formátů skrývá.

Bez produktů Microsoft Office se většina kanceláří v současnosti zřejmě neobejde. Dle mého názoru by však měl být otevřený formát ODF podporován například ve školství. S lepšími vyhlídkami do budoucna a nižší cenou totiž dokáže nabídnout stejné funkce, jako formát MS Office.

7. Seznam použitých zdrojů

- [1] BRADLEY, Neil. XML Companion, Addison-Wesley, Harlow 2001, 864s. ISBN 0-201-34285-5.
- [2] GRUSOVÁ, Lucie. XML pro úplné začátečníky, ComputerPress, Brno 2002, 196s. ISBN 80-7226-697-7.
- [3] KOSEK, Jiří. XML pro každého. Grada Publishing, Praha 2000, 164s. ISBN 80-7169-860-1.
- [4] LAURENT, Simon, FITZGERALD, Michael. XML Pocket Reference, Third Edition. O'Reilly Media, 2009. 176s. ISBN 978-0-596-10050-6.
- [5] MLÝNKOVÁ, Irena, et al. Technologie XML – Principy a aplikace v praxi. Grada Publishing, Praha 2008. 272s. ISBN 978-80-247-2725-7.
- [6] SKONNARD, Aaron, GUDGIN Martin. XML pohotová referenční příručka. ComputerPress, Brno 2005. 344s. ISBN 80-247-0972-4
- [7] JELÍNEK, L. Opendocument - otevřený formát pro vaše kancelářské dokumenty [Online].[Cit. 8.2.2011].
<<http://www.opendocument.cz>>
- [8] KOSEK, J. Domovská stránka Jirky Koska [Online]. [Cit. 23.3.2011].
<<http://www.kosek.cz>>
- [9] KOŠATA, B. Seriál XML [Online]. [Cit. 5.4.2011].
<<http://www.root.cz/serialy/xml-extensible-markup-language>>
- [10] Extensible Markup Language [Online]. [Cit. 4.3.2011].
<<http://www.w3.org/XML>>

- [11] OASIS [Online]. [Cit. 15.11.2010].
<<http://www.oasis-open.org/home/index.php>>
- [12] OpenDocument Format Alliance [Online]. [Cit. 5.3.2011].
<<http://www.odfalliance.org>>
- [13] OpenDocument- Wikipedie [Online].[Cit. 16.2.2011].
<<http://cs.wikipedia.org/wiki/OpenDocument>>
- [14] OpenOffice.org - The Free and Open productivitz Suite [Online].
<<http://www.openoffice.org>>
- [15] Software602 [Online].[Cit.2.4.2011].
<<http://www.602.cz>>
- [16] Společnost pro vyzkum a podporu open-source [Online].[Cit. 25.11.2010].
<<http://www.oss.cz>>
- [17] XPath Language [Online]. [Cit. 22.2.2011].
<<http://www.w3.org/TR/xpath>>
- [18] XPath - Wikipedie [Online]. [Cit. 20.2.2011].
<<http://cs.wikipedia.org/wiki/XPath>>
- [19] World Wide Web Consortium (W3C) [Online]. [Cit 23.2.2011].
<<http://www.w3.org>>

7.1 Seznam obrázků

Obrázek č. 1- Osy	24
< http://www.kosek.cz/xml/xslt/pic/model-axis.png >	
Obrázek č. 2- Zpracování XSL	30
< http://www.kosek.cz/xml/xslt/pic/model-axis.png >	
Obrázek č. 3- dp1.html	53
Obrázek č. 4- menudp1.html.....	54
Obrázek č. 5- Seznam.html.....	54
Obrázek č. 6- DP1.fo	60
Obrázek č. 7-Vzhled	62
Obrázek č. 8- Datová struktura	63
Obrázek č. 9- Formulářové prvky.....	64
Obrázek č. 10- Titulní strana	65
Obrázek č. 11- Změna odběratele	68
Obrázek č. 12- Změna zasílací adresy	69
Obrázek č. 13- Používané editory.....	71
Obrázek č. 14- Zobrazený obsah	74

7.2 Seznam tabulek

Tabulka č. 1- Ukázky XPath.....	26
Tabulka č. 2- Druhy souborů	36
Tabulka č. 3- Historie ODF	39
Tabulka č. 4- Možnosti otevírání.....	73

8. Přílohy

Seznam příloh:

Příloha č. 1- Dotazník	1
Příloha č. 2- Naskenovaný formulář firmy PRE	3

Další přílohy jsou dostupné na přiloženém CD.

Příloha č. 1- Dotazník

Dotazník byl vytvořen pomocí aplikace Google Docs, u většiny otázek bylo nutno odpověď zaškrtnout, čímž se zobrazila příslušná další otázka. Výsledky dotazníkového průzkumu jsou k dispozici na CD, v souboru dotazník.ods.

1) Jaký textový editor nejčastěji používáte?

- a) MS Word 2007 nebo 2010 -> skok na otázku 2-1
- b) MS Word 2003 nebo XP -> skok na otázku 2-1
- c) MS Word, jinou nebo neznámou verzi -> skok na otázku 2-1
- d) OpenOffice.org Writer -> skok na otázku 3
- e) jiný -> skok na otázku 4-1

2-1) Využíváte legální verzi MS Office

- a) ANO -> skok na otázku 2-2
- b) NE -> skok na otázku 2-2
- c) Nevím -> skok na otázku 2-2

2-2) Hlavní důvod využívání MS Office

- a) Nabízi funkce, které nemá konkurence -> skok na otázku 5
- b) Dokumenty Office brány jako standard -> závěr dotazníku
- c) Nemám na výběr -> závěr dotazníku
- d) jiný (vypisovací možnost) -> závěr dotazníku

3) Hlavní důvod, proč používáte OpenOffice.org

- a) Je to zdarma -> závěr dotazníku
- b) Nabízi funkce, které nemá konkurence -> skok na otázku 5
- c) Dokumenty ODF brány jako standard -> závěr dotazníku
- d) Nemám na výběr -> závěr dotazníku
- e) jiný (vypisovací možnost) -> závěr dotazníku

4-1) Jaký používáte textový editor?

(vypisovací otázka)

4-2) Hlavní důvod, proč používáte tento editor

- a) Je to zdarma -> závěr dotazníku
- b) Nabízi funkce, které nemá konkurence -> skok na otázku 5
- c) Nemám na výběr -> závěr dotazníku
- d) jiný (vypisovací možnost) -> závěr dotazníku

5) Které funkce nabízí Váš editor oproti konkurenci navíc?

Textová otázka -> závěr dotazníku

Příloha č. 2- Naskenovaný formulář firmy PRE

PRE
Žádost o změnu / uzavření / ukončení smlouvy

Důvod žádosti:

<input type="checkbox"/> Změna odběratele (Převod) ¹⁾ <i>(Nový odběratel vyplňuje část 2., 6., 7., stávající odběratel část 1., 2., 3.)</i>	<input type="checkbox"/> Ukončení odběru ²⁾ <i>(Vyplňte část 1., 2., 3. a 4.)</i>	<input type="checkbox"/> Změna platebního způsobu <i>(Vyplňte část 1. a 5.)</i>
<input type="checkbox"/> Elektronická fakturace ³⁾ <i>(Vyplňte část 1. a 5.)</i>	<input type="checkbox"/> Změna ostatních údajů <i>(Vyplňte část 1. a 5.)</i>	<input type="checkbox"/> Změna zasilací adresy <i>(Vyplňte část 1. a 3.)</i>

1. Identifikace stávajícího odběratele a odběrného místa

Jméno a příjmení / název / firma datum narození IČ

Číslo odběrného místa (naleznete na faktuře) 8 1 1 tel. / e-mail

Číslo zákaznického účtu (naleznete na faktuře)
 adresa odběrného místa

2. Údaje o měřicím zařízení

Číslo elektroměru Stav elektroměru uvedený odběratelem³⁾ T1 T2

3. Adresa pro zaslání faktur a korespondence *(V případě Ukončení odběru a při Převodu bude adresa použita pro zaslání konečné faktury)*

Jméno a příjmení / název / firma datum narození:

ulice č. orientační č. popisné

obvod obec PSČ

číslo bankovního účtu k vyrovnání závazků:

4. Důvod ukončení odběru

Stěhování / Ukončení činnosti Jiné

5. Jiné změny

Popis

Stávající odběratel (v případě Ukončení odběru a/nebo Převodu odběru) zároveň žádá společnost PREdistribuce, a. s., aby ke dni ukončení odběru elektřiny v odběrném místě dle této žádosti byla ukončena účinnost smlouvy o připojení pro toto odběrné místo. Pražská energetika, a. s., se zavazuje předat uvedenou žádost společnosti PREdistribuce, a. s., IČ 27376516. Prohlašuji, že mnou uvedené údaje jsou pravdivé a souhlasím s nimi, což stvrzuji vlastnoručním podpisem.

Podpis stávajícího odběratele: v dne

Vyvětlivky

- 1) Převodem se rozumí změna odběratele v odběrném místě specifikovaném v této žádosti bez přerušení dodávky elektřiny a bez změny charakteru odběru. Stávající odběratel na základě této žádosti o ukončení dodávky elektřiny končí odběr elektřiny v odběrném místě. Místo něho bude v odběrném místě specifikovaném v této žádosti odebrat elektřinu od dodavatele Pražské energetiky, a. s., nový odběratel na základě této žádosti o uzavření smlouvy o sružených službách dodávky elektřiny. V případě, že nebude zahájena dodávka elektřiny novému odběrateli, dojde na základě této žádosti pouze k ukončení dodávky elektřiny stávajícímu odběrateli do 30 dnů od doručení žádosti.
- 2) K ukončení smlouvy na dodávku elektřiny z důvodu faktického ukončení odběru zákazníkem dojde provedením konečného odečtu stavu elektroměru do 30 dnů od doručení této žádosti. Dodavatel následně vystaví zákazníkovi konečnou fakturu. Není-li elektroměr přístupný, nemůže být odběr ukončen, a je proto v zájmu zákazníka dohodnout termín návštěvy a zajistit zpřístupnění elektroměru. I v případě, že je elektroměr přístupný, doporučujeme zákazníkům, aby byli konečnému odečtu přítomni. Vzájemné závazky budou vyrovnány v režimu stávající platební metody, pokud nebude dohodnuto jinak.
- 3) V případě Převodu bude Pražskou energetikou, a. s., tento stav měřicího zařízení respektován jako počáteční stav zahájení dodávky elektřiny novému odběrateli, pokud nebude při odečtu provedeným příslušným provozovatelem distribuční soustavy zjištěn odlišný stav.

Zákaznické e-centrum PRE
www.pre.cz/e-centrum

Zákaznická linka PRE
Tel.: 267 055 555
Fax: 267 055 505
E-mail: pre@pre.cz
Provozní doba: Po–Pá 7.00–19.00
Záznamník: 19.00–7.00

Zákaznické centrum PRE
Praha 1, Jungmannova 31 (palác Adria)
Praha 4, Vladimírova 18
Praha 9, Ocelářská 5a
Provozní doba: Po–Čt 9.00–18.00
Pá 9.00–12.00

Centrum energetického poradenství PRE
Jungmannova 28 (Palác TeTa), Praha 1
Provozní doba: Po–Pá 10.00–18.00
Tel.: 267 055 555
E-mail: poradce@pre.cz
www.energetickyporadce.cz

Adresa pro doručování: Pražská energetika, a. s., Na Hroudě 1492/4, 100 05 Praha 10 otočit na druhou stranu

OBCHODNÍ TISKÁRNA, s. r. o., KOLÍN PRE 049 / 10.10

6. Nový odběratel

Jméno a příjmení / název / firma

trvalé bydliště / sídlo / ulice

č. orientační⁴⁾

č. popisné

/ místo podnikání

obvod

obec

PSČ

datum narození

IČ

DIČ

Jednatel / zástupce: ⁵⁾

Kontakty:

telefon (mobil)

fax

e-mail

Požadovaná distribuční sazba ⁶⁾

účel odběru

odběr v kategorii D dle příslušného právního předpisu („domácnosti“)

odběr v kategorii C dle příslušného právního předpisu („odběratel, který není domácností“)

Elektronická fakturace⁷⁾: e-mail

Zasílací adresa

odběrné místo

trvalé bydliště / sídlo / místo podnikání

jiné (vyplňte následující pole Adresa pro zasílání faktur a korespondence)

Adresa pro zasílání faktur a korespondence:

7. Platební způsob

Bezhotovostně dle vlastního výběru na účet dodavatele (prostřednictvím banky, pošty atd.). Požadujete-li některý z dále uvedených způsobů, označte ho křížkem a vyplňte příslušné údaje.

Příkaz k úhradě

Přímé inkaso (vyplňte č. účtu)

O tomto způsobu platby informujte svou banku

SIPO (vyplňte spoj. č. SIPO)

Pouze pro zálohy

Frekvence záloh ⁸⁾

- měsíčně
 dvouměsíčně
 čtvrtletně
 pololetně
 ročně

číslo účtu / kód banky

spojové číslo SIPO

V případě Převodu (tj. změna odběratele v odběrném místě dle této žádosti) uskutečněného prostřednictvím písemného styku platí následující podmínky:

Na základě této žádosti (návrhu na uzavření smlouvy) je uzavřena Smlouva o sdružených službách dodávky elektřiny (dále jen „Smlouva“) pro odběrné místo specifikované v této žádosti. Smlouvu uzavírá Pražská energetika, a.s., (dále také „dodavatel“) s novým odběratelem (dále také „zákazník“) podle zákona č. 458/2000 Sb., energetický zákon, a jeho prováděcích právních předpisů, v režimu zák. č. 513/1991 Sb., obchodní zákoník, za následujících podmínek:

I. Předmětem plnění Smlouvy je za podmínek v ní dohodnutých

- dodávka sjednaného množství elektřiny mezi dodavatelem a zákazníkem v odběrném místě zákazníka dle Smlouvy,
- zajištění dopravy a souvisejících služeb dodávky sjednaného množství elektřiny dodavatelem pro zákazníka,
- převzetí závazku zákazníka odebrat elektřinu z elektrizační soustavy dodavatelem spolu s odpovědností za odchylku vztahující se k odběrnému místu dle Smlouvy,
- úhrada smluvní ceny za dodávky silové elektřiny dle této Smlouvy a úhrada regulovaných cen zákazníkem dodavatel.

II. Podmínky dodávky elektřiny

Pro dodávky elektřiny dle Smlouvy platí Obchodní podmínky sdružených služeb dodávky elektřiny č. SNN 03/2009 vydané dodavatelem (dále jen „OPD“), které jsou její nedílnou součástí. Způsob stanovení výše zálohy je stejný jako u stávajícího odběratele.

III. Specifikace odběrného místa zákazníka

Rezervovaná kapacita (hodnota hlavního jističe před elektroměrem) i ostatní parametry v odběrném místě jsou ve stejné výši jako u stávajícího odběratele v době převodu dodávky elektřiny.

IV. Cena dodávky elektřiny

- Cena za silovou elektřinu je smluvní a je sjednána ve formě cenového tarifu („Produktu silové elektřiny“) určeného v bodě b) tohoto článku a specifikovaného v Ceníku Pražské energetiky, a.s., platném pro nového odběratele v době převodu dodávky (dále jen „Ceník“). Ceník je nedílnou součástí Smlouvy.⁹⁾ Regulované ceny jsou stanoveny ve výši podle platného cenového rozhodnutí Energetického regulačního úřadu. Cena za distribuci je sjednána v distribuční sazbě uvedené v bodě b) tohoto článku.
- Smluvní strany sjednávají Produkt silové elektřiny uvedený v Ceníku pro požadovanou distribuční sazbu a distribuční sazbu dle požadavku nového odběratele uvedeného v této žádosti.

V. Trvání Smlouvy

- Smlouva nabývá platnosti a účinnosti dnem doručení souhlasu dodavatele s Převodem na základě této žádosti zákazníkovy. Smlouva je uzavřena na dobu neurčitou.
- Smlouvu lze jednostranně vypovědět kteroukoliv smluvní stranou. Výpovědní lhůta je 3 měsíce a počíná běžet po doručení písemné výpovědi druhé smluvní straně.
- Další způsoby a podmínky ukončení této Smlouvy jsou stanoveny v OPD.

Prohlašuji, že mnou uvedené údaje jsou pravdivé a souhlasím s uzavřením Smlouvy za výše uvedených podmínek, což stvrzuji vlastnoručným podpisem. Současně žádám společnost PREdistribuce, a. s., o uzavření smlouvy o připojení pro předmětné odběrné místo za stávajících podmínek. Pražská energetika, a. s., se zavazuje předat uvedenou žádost společnosti PREdistribuce, a. s., IČ 27376516.

Nový odběratel (pouze u Převodu):

(podpis nového odběratele či jeho zástupce / zástupců)

Převzal zástupce Pražské energetiky, a. s.,

dne

Vysvětlivky

4) U nových staveb, které ještě nemají přiřazeno č. orientační a popisné, uveďte prosím číslo katastru a parcely.

5) V případě, že jménem nového odběratele jedná při podávání této žádosti zástupce či jedná statutárním orgánem, vyplní se jméno a příjmení jednatel či zastupující osoby/osob a důvod jednání či zastoupení (tj. např. člen představenstva, zástupce dle plné moci, vedoucí oddělení). V případě zastoupení dle plné moci doloží zástupce k této žádosti plnou moc.

6) a) V případě, že nový odběratel nevyplní požadovanou distribuční sazbu dle příslušného cenového rozhodnutí ERÚ, požaduje stejnou distribuční sazbu, která je u stávajícího odběratele v době převodu dodávky elektřiny. V případě, že není vyplněná požadovaná sazba u převodu dodávky elektřiny mezi odběratelem zařazeným v kategorii C dle příslušného právního předpisu („odběratel, který nejsou domácností“) a odběratelem, který je v kategorii D dle příslušného právního předpisu („domácnosti“), požaduje nový odběratel distribuční sazbu, která charakterem odpovídá distribuční sazbě u stávajícího odběratele v době převodu dodávky elektřiny.

b) V případě, že nový odběratel nebude v době převodu dodávky splňovat podmínky či nedoloží splnění podmínek pro přidělení požadované distribuční sazby, požaduje přidělení distribuční sazby dle postupu pro případ nevyplnění požadavku distribuční sazby uvedeného v odst. a) této vysvětlivky.

7) V případě požadavku na vystavování faktur v elektronické podobě není zasílána faktura v písemné podobě.

8) Pokud nový odběratel nevybere příslušnou frekvenci záloh, požaduje frekvenci záloh, která je u stávajícího odběratele v době převodu dodávky elektřiny.

9) Ceník je veřejně přístupný na internetových stránkách Pražské energetiky, a. s., či v jejím Zákaznickém centru PRE a nový odběratel podpisem této žádosti prohlašuje, že je mu tento Ceník znám.

Upozornění: V případě chybných nebo nepravdivých údajů nemusí vzhledem k jejich důležitosti a charakteru dojít ke zpracování Vašeho požadavku. Pokud nastane tato situace, budeme Vás písemně nebo telefonicky kontaktovat. Proto ve svém vlastním zájmu uvádějte platné kontaktní údaje (adresa, telefon). Doporučujeme Vám seznámit se s Obchodními podmínkami dodávky elektřiny, které jsou k dispozici v našem Zákaznickém centru PRE, na internetových stránkách (www.pre.cz) a na Zákaznické lince PRE. V případě jakýchkoli problémů nebo nejasností při vyplňování této žádosti se laskavě obraťte na naše zástupce v Zákaznickém centru PRE nebo telefonicky na Zákaznickou linku PRE.