

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

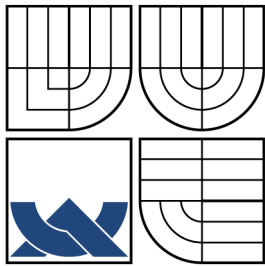
IMPLEMENTACE STANDARDU BIBLIOGRAFICKÉ CITACE
BIBTEX POMOCÍ TECHNOLOGIE XML

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

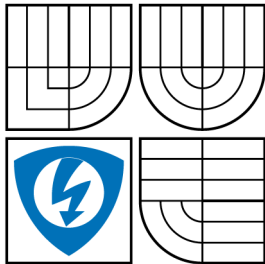
AUTOR PRÁCE
AUTHOR

BC. LADISLAV MAZAČ

BRNO 2010



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY
A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV TELEKOMUNIKACÍ



FACULTY OF ELECTRICAL ENGINEERING AND
COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

IMPLEMENTACE STANDARDU BIBLIOGRAFICKÉ CITACE BIBTEX POMOCÍ TECHNOLOGIE XML

IMPLEMENTATION OF A STANDARD OF BIBLIOGRAPHIC QUOTATION BIBTEX
USING THE XML TECHNOLOGY

DIPLOMOVÁ PRÁCE
MASTER'S THESIS

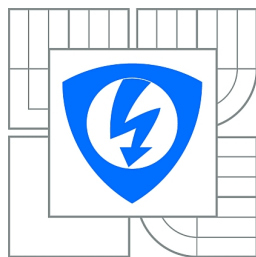
AUTOR PRÁCE
AUTHOR

BC. LADISLAV MAZAČ

VEDOUCÍ PRÁCE
SUPERVISOR

MGR. PAVEL RAJMIC, PH.D.

BRNO 2010



VYSOKÉ UČENÍ
TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

Ústav telekomunikací

Diplomová práce

magisterský navazující studijní obor
Telekomunikační a informační technika

Student: Bc. Ladislav Mazač

ID: 88552

Ročník: 2

Akademický rok: 2009/2010

NÁZEV TÉMATU:

Implementace standardu pro bibliografické citace BibTeX pomocí technologie XML

POKYNY PRO VYPRACOVÁNÍ:

Na základě navrženého DTD pro databázi XML, která bude obsahovat několik publikací různých typů (knihy, články v časopisech, webové stránky, výzkumné zprávy apod.), implementujte XSL schéma, pomocí kterého umožníte bezchybnou konverzi databáze z XML do BibTeX.

DOPORUČENÁ LITERATURA:

[1] GRUSOVÁ, L.: XML pro úplné začátečníky. Computer Press, Praha, 2002. ISBN 80-7226-697-7

[2] Kol.: XML technologie: principy a aplikace v praxi. Grada Publishing, Praha, 2008. ISBN 978-80-247-2725-7

[3] RYBIČKA, J.: LaTeX pro začátečníky (3. vydání). Konvoj, Brno, 2003. ISBN 80-7302-049-1

[4] GOOSENS, M.: The LaTeX Companion, Addison Wesley, ISBN 0201541998

Termín zadání: 29.1.2010

Termín odevzdání: 26.5.2010

Vedoucí práce: Mgr. Pavel Rajmic, Ph.D.

prof. Ing. Kamil Vrba, CSc.

Předseda oborové rady

UPOZORNĚNÍ:

Autor diplomové práce nesmí při vytváření diplomové práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

ABSTRAKT

Tato diplomová práce se zabývá implementací standardu bibliografických citací $\text{BIB}\text{T}\text{E}\text{X}$ s pomocí technologie XML. Součástí této práce je teoretický úvod do problematiky XML technologií. Další částí je seznámení se sázecím programem $\text{L}\text{A}\text{T}\text{E}\text{X}$ a jeho programem $\text{BIB}\text{T}\text{E}\text{X}$ a obecně s jeho funkcí. Dále je znázorněn samotný princip implementace a podstatné části zdrojového kódu XSL transformací, a to nejen pro výstupní *.bib soubory, ale i pro soubory (*.tex, *.html). Ověření funkčnosti implementace je demonstrováno využitím $\text{BIB}\text{T}\text{E}\text{X}$ ové databáze použité literatury, vygenerované pomocí XML technologie v této práci.

KLÍČOVÁ SLOVA

XML technologie, XSL, DTD, $\text{L}\text{A}\text{T}\text{E}\text{X}$, $\text{BIB}\text{T}\text{E}\text{X}$, impelementace, bibliografická citace, tex, html

ABSTRACT

This master's thesis considers implementation of a standard of bibliographic quotation BibTeX using the XML technology. This project starts with theoretical introduction into XML technology problematics. Next part contains introduction into $\text{L}\text{A}\text{T}\text{E}\text{X}$ and its subprogram $\text{BIB}\text{T}\text{E}\text{X}$ and into its function. Later, the principle of implementation and important parts of XSL transformation (from output files *.bib. *.tex, *.html) source code will be described. Verification of functionality of the implementation is demonstrated using $\text{BIB}\text{T}\text{E}\text{X}$ database of used literature generated by XML technology.

KEYWORDS

XML technology, XSL, DTD, $\text{L}\text{A}\text{T}\text{E}\text{X}$, $\text{BIB}\text{T}\text{E}\text{X}$, implementation, bibliographical citation, tex, html

MAZAČ L. *Implementace standardu bibliografické citace BIB_TE_X pomocí technologie XML*. Brno: Vysoké učení technické v Brně. Fakulta elektrotechniky a komunikačních technologií. Ústav telekomunikací, 2010. Počet stran 61. Diplomová práce. Mgr. Pavel Rajmic Ph.D.

PROHLÁŠENÍ

Prohlašuji, že svou diplomovou práci na téma „Implementace standardu bibliografické citace BibTeX pomocí technologie XML“ jsem vypracoval samostatně pod vedením vedoucího diplomové práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením této diplomové práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení § 152 trestního zákona č. 140/1961 Sb.

V Brně dne

.....

(podpis autora)

Chtěl bych poděkovat Mgr. Pavlu Rajmicovi Ph.D, vedoucímu mé diplomové práce, za odborné vedení, cenné rady a připomínky, které mi při tvorbě mé diplomové práce ochotně poskytoval. Dále chci poděkovat mé rodině a všem blízkým, kteří mě v mém studiu podporovali.

OBSAH

Úvod	11
1 Teoretická část	12
1.1 Bibliografické citace	12
1.2 XML	13
1.2.1 Obecně o XML	13
1.2.2 DTD	16
1.2.3 Jmenné prostory	18
1.2.4 Jazyk XPath	19
1.2.5 XSL	21
1.3 BiB _{TEX}	25
1.3.1 Ve zkratce o programu L _A T _E X a BiB _{TEX}	26
1.3.2 Aplikace programu BiB _{TEX}	26
2 Implementace citací pomocí XML technologií	29
2.1 Proces implementace	29
2.2 Zdrojové soubory	31
2.2.1 DTD dokument	31
2.2.2 XML dokument	32
2.2.3 XSL dokument pro výstupní soubor bib	34
2.2.4 XSL dokument pro výstupní soubor tex	40
2.2.5 XSL dokument pro výstupní soubor html	43
2.3 Ukázky výstupních souborů	47
2.3.1 Ukázka výstupu souboru *.bib	47
2.3.2 Ukázka výstupu souboru *.tex převedeného na soubor *.pdf	47
2.3.3 Ukázka výstupu souboru *.html	47
2.4 Grafický program pro práci s parserem Saxon	53
2.4.1 Základní práce v programu a jeho popis	54
3 Závěr	56
Literatura	57
Seznam symbolů, veličin a zkratk	59
Seznam příloh	60

A	Popis přiloženého datového média	61
A.1	Zdrojové soubory potřebné pro transformace	61
A.2	Elektronická podoba diplomové práce	61
A.3	Zdrojové kódy ke grafickému programu k obsluze parseru Saxon9he .	61

SEZNAM OBRÁZKŮ

1.1	Jazyk XPath–ukázka XML stromu spolu s uzly	20
1.2	Transformace pomocí XSLT	22
2.1	Proces implementace citací s využitím XML technologií a grafického rozhraní	30
2.2	Ukázka výstupního bib souboru použitého v diplomové práci	48
2.3	Ukázka výstupního pdf souboru vygenerovaného ze souboru tex se všemi typy citací	49
2.4	Ukázka výstupního pdf souboru vygenerovaného ze souboru tex pro citované zdroje v této diplomové práci	50
2.5	Ukázka výstupního html souboru se všemi typy citací	51
2.6	Ukázka výstupního html souboru pro citované zdroje v této diplomové práci	52
2.7	Ukázka výstupního html souboru	53
2.8	Ukázka výběru transformčního předpisu	55

SEZNAM TABULEK

1.1	Vyjádření obsahu elementy operátory četnosti	17
1.2	Příklady XPath dotazů	21
1.3	XSLT - Vybrané atributy elementu xsl:output	24
1.4	Atributy elementu xsl:sort	25
1.5	Styly pro formátování referencí programu BIB _T E _X	26
2.1	Atributy elementu xsl:stylesheet pro výstupní *.bib soubor	35
2.2	Atributy elementu xsl:output pro výstupní *.bib soubor	35
2.3	Číselné entity, formátovací znaky a jejich popis	38
2.4	Atributy elementu xsl:stylesheet pro výstupní *.tex soubor	40
2.5	Atributy elementu xsl:output pro výstupní *.tex soubor	40
2.6	Použité příkazy sázecího programu L _A T _E X	41
2.7	Atributy elementu xsl:stylesheet pro výstupní *.tex soubor	43
2.8	Atributy elementu xsl:output pro výstupní *.tex soubor	43
2.9	Ladící a chybové hlášení programu	55

ÚVOD

V informačních a komunikačních technologiích a v mnoha dalších oborech je možné stále častěji zaslechnout termín XML (eXtensible Markup Language). Český ekvivalent můžeme nalézt například pod spojením slov rozšířitelný značkovací jazyk. Jedná se o jazyk, který využívá značek neboli takzvaných tagů. Tento jazyk se v současné době používá pro tvorbu internetových stránek, pro generování výstupu z různých aplikací, případně jako rozhraní mezi informačními systémy. Lze tedy říci, že jazyk XML slouží k interpretaci textů (dat).

V této diplomové práci je řešena implementace standardu bibliografických citací $\text{BIB}_{\text{T}}\text{E}_{\text{X}}$ s využitím technologie XML. $\text{BIB}_{\text{T}}\text{E}_{\text{X}}$ je program, který využívá sázecí program $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ při tvorbě textů pro usnadnění práce s bibliografickými citacemi. Program $\text{BIB}_{\text{T}}\text{E}_{\text{X}}$ uchovává citovanou literaturu v souborech s příponou `bib`. Cílem bylo získat plnohodnotný soubor bibliografických citací s příponou `bib`, máme-li k dispozici použitou literaturu v souboru s příponou `xml`. XML transformace vytvořené v této diplomové práci umožňují vytvořit `bib` soubor, který je plnohodnotně využitelný pro citování zdrojů při psaní textu v programu $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$. Také jsou vytvořeny transformace pro zobrazení seznamu zdrojů obsažených v databázi bibliografických citací. Tyto transformace tvoří seznamy ve formátu `tex` a `html`. Součástí práce je také aplikace, která je napsaná v `.NET` a slouží pro grafickou obsluhu parseru Saxon a pro generování souborů `pdf` ze souborů `tex`.

Tento dokument se dělí na dvě základní části. První část objasňuje teorii týkající se technologie XML a způsoby využití programu $\text{BIB}_{\text{T}}\text{E}_{\text{X}}$. V druhé části je vysvětleno řešení požadované implementace, popsání struktury souboru DTD (Document Type Definition) a XSL (eXtensible Stylesheet Language) šablon pro převod souboru do již zmiňovaných výstupních formátů. Krátká část je také věnována grafickému prostředí na obsluhu parseru Saxon. Jedná se spíše o metodiku práce s tímto programem, než o přiblížení vnitřní struktury programového kódu.

1 TEORETICKÁ ČÁST

Teoretická oblast se dělí na několik částí. Hned ze začátku této kapitoly je stručně popsána citace a její využití. V druhé části jsou vysvětleny základy technologie XML a poslední část objasňuje funkčnost programu `BIBTEX` a přináší základní informace o sázecím programu `LATEX`.

1.1 Bibliografické citace

V rámci každého odborného textu by měly být uvedeny takzvané bibliografické citace. Uvádí se z důvodu zachycení myšlenkové vazby mezi originálním textem a čerpanými myšlenkami z jiných zdrojů. Nejpodstatnější důvody pro citování zdrojů jsou následující:

- Prokázání vlastní znalosti tématu;
- dodržení autorské etikety a zákonu o autorských právech;
- odkázání čtenářů na další zdroje.

Citování podléhá citačním normám. Citace knihy může vypadat následovně:

Grusová, L.: XML pro úplné začátečníky. Praha: Computer Press, vydání první vydání, 2002, ISBN 80-7226-697-7.

Toto je ukázka konkrétní citace, která je použita v této práci. Dále se v textu většinou v samotném závěru používá soupis bibliografických citací, který je běžně uspořádán abecedně podle prvního autora nebo v numerické posloupnosti, která odpovídá pořadí použité literatury v textu. V příslušném místě se vloží označení citace a je vysázena jako číslo uvozené hranatými závorkami, které odpovídá soupisu bibliografických citací [7].

Samotná metodika citování je určena jednotlivými normami. V rámci České republiky se využívají normy ČSN 690 a ČSN 690-2. Blíže jsou tyto normy popsány v literatuře [2] a [3]. Existují různé druhy citací. Mezi základní typy citací by mohly být řazeny následující:

- Kniha;
- článek;
- internetový zdroj;
- manuál;

- brožura;
- standard.

V normách jsou určeny povinné a nepovinné atributy citací a také jejich konkrétní řazení, které je samozřejmě závislé na typu citace. Povinnost atributu ovšem neznamená, že musí být u každé citace uveden. Příkladem může být například typ citace internetový zdroj. Pokud autor neuvede například datum poslední aktualizace stránky, nelze jej uvést v citaci. I když tedy mluvíme o povinných atributech, programy pro tvorbu databáze bibliografických citací jako je JabRef, případně transformace pomocí XML technologií vyvinuté v rámci této diplomové práce, umožní uložení konkrétní citace bez obsahu všech povinných atributů, které jmenují normy.

Je tedy záměrem neomezovat pevně atributy databáze bibliografických citací. Zmenšovala by se tak využitelnost podpůrných programů, transformací na konkrétní normu citování. Normy citování se mohou lišit.

1.2 XML

Tato kapitola si klade za cíl pojednat o základech jazyka XML. Čtenáři bude vysvětleno využití DTD¹ a XSL souborů. Tento základ je nezbytný pro pochopení samotné implementace standardu bibliografických citací BIB_TE_X pomocí technologie XML.

1.2.1 Obecně o XML

Zkratka XML znamená eXtensible Markup Language, tedy v českém překladu rozšiřitelný značkovací jazyk, patřící do skupiny značkovacích jazyků [7]. Tyto jazyky využívají k označování textů již zmiňované značky, kterým se mimo jiné říká tagy. XML není prvním značkovacím jazykem. Předchůdcem tohoto dnes hojně využívaného značkovacího jazyka je jazyk SGML (Standard Generalized Markup Language). Tento jazyk byl však poměrně složitý a proto nenašel moc velké uplatnění. Položil však základ hojně využívanému HTML (HyperText Markup Language) a XML jazyku, kterému se budeme dále v textu věnovat.

Jazyk XML je standardizovaný konsorciem W3C, přispěl k vývoji souvisejících metod, které se dnes souhrně nazývají XML technologie. Slovo technologie označuje soubor postupů, nástrojů a procesů sloužících k nějaké činnosti. V případě XML může jít nejen o transfer dokumentů nebo jejich transformace atd., ale i o jejich ukládání v databázi a zpracování databázovým způsobem. Značkovací jazyk XML

¹DTD není v dnešních aplikacích již tak hojně využívanou záležitostí. Daleko více využívaným prvkem pro vytovření sady pravidel pro tvorbu dokumentu jsou takzvané XSD soubory. Více informací je v kapitole o DTD

slouží jako výměnný formát dat, tvoří syntaktickou základnu sémantického webu, je na něm založena servisně orientovaná integrace, lze jej dokonce považovat za nový databázový model. Existence XML vedla k rozvoji dalších jazyků a podpůrných prostředků [6]. Některé z mála podpůrných jazyků si představíme dále v této práci a to v kapitole 1.2.4.

Jazyk XML má svá syntaktická pravidla, která musí být striktně dodržována. Základní stavební prvek XML souboru je takzvaný element. Je uvozen počáteční značkou a ukončen je takzvanou koncovou značkou, které jsou zapsány do lomených závorek (<>). Koncová značka má před názvem elementu umístěno lomítko. Těmto značkám se také říká párové značky. Text umístěný mezi značkami je samotný obsah elementu. Zapsaný element tedy může vypadat následovně:

```
<type>book</type>
```

Obsah elementu je tedy slovo book, může se stát, že element bude prázdný. V tomto případě se dá napsat do následujícího tagu:

```
<type/>
```

Při výběru názvu elementů má uživatel téměř neomezené možnosti, ovšem musí dodržet některé základní pravidla tvorby XML dokumentu jako jsou [7]:

- XML musí vždy obsahovat kořenový element;
- zmiňovaný jazyk je case-sensitive, tedy rozlišuje malá a velká písmena;
- elementy se nesmí křížit. Pokud nějaký element obsahuje počáteční značku jiného elementu, musí obsahovat i jeho koncovou značku;
- u každého elementu může existovat atribut, který nese rozšiřující údaje o daném elementu. Zápis může vypadat následovně:

```
<type sequence="hodnota_atributu_např._číslo_1">book</type>
```

Atributy se oddělují mezerou od názvu elementu, proto nesmí název elementu obsahovat mezeru;

- XML nesmí obsahovat zakázané znaky. Musí být nahrazeny vestavěnými znakovými entitami:
 - Znak & nahrazuje entita & ;
 - znak " nahrazuje entita " ;
 - znak ' nahrazuje entita ' ;

- znak < nahrazuje entita < ;
 - znak > nahrazuje entita > .
- všechny názvy elementů musí začínat písmenem nebo podtržítkem. Následujícími znaky ve jménu elementu mohou být písmena, číslice, pomlčky, tečky a podtržítka.

Na XML dokument lze pohlížet jako na stromovou strukturu. Jak je již známo, dokument musí obsahovat jeden kořenový element, který se dál může větvit. Lépe viditelné to bude na následujícím příkladě²:

```
<bibliography>
  <unit>
    <type>book</type>
    .
    .
    .
    <year>2002</year>
    <isbn>Číslo knihy ISBN</isbn>
  </unit>
</bibliography>
```

Kořenovým elementem je v tomto případě element s názvem `bibliography`, který je zároveň rodičem elementu `unit`. Potomky elementu `unit` jsou všechny elementy v něm obsažené. Elementy `type`, `year`, `isbn` jsou ve vztahu k sobě sourozenci. Získání správné představy o struktuře XML je velmi důležité pro pochopení jazyka XPath, který je velmi důležitou součástí XML technologie. Jeho bližší význam bude přiblížen v kapitole 1.2.5.

Komentáře se zapisují následovně:

```
<!--Komentář v XML technologiích-->
```

Každý XML dokument musí obsahovat takzvanou deklaraci, která by měla splňovat následující podmínky [7]:

- v deklaraci se musí používat pouze malá písmena;
- deklarace začíná znaky `<?` a končí sekvencí `?>`;
- jako první je v deklaraci použito slovo `xml`;

²Pokud budou dále v textu uváděny ukázky zápisů zdrojových textů, tak právě s ohledem na tuto strukturu, která je základem XML souboru pro implementaci bibliografických citací.

- v deklaraci není možno měnit pořadí atributů. Jediné správné pořadí je tedy version a poté encoding;
- hodnoty atributu musí být jako standardně označeny uvozovkami nebo apostrofy. Tyto dvě možnosti se nesmí u jedné hodnoty zaměňovat.

Příklad zápisu deklarace XML dokumentu:

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

Ukázková deklarace říká, že byla použita XML technologie verze 1.0 Splňuje-li dokument všechny v této kapitole uvedené požadavky, říkáme, že je well-formed, nebo-li správně strukturovaný. Toto je nutná podmínka pro korektnost XML dokumentů.

1.2.2 DTD

Document Type Definition je v českém překladu definice typu dokumentu, kterým se stanoví gramatika přístupné třídy dokumentu. Dalším užívaným názvem je schéma dokumentu [6]. Pokud dokument XML je well-formed a odpovídá také svému schématu, můžeme říct, že je také validní. V tomto případě jsou tedy splněny dva základní prvky pro korektnost XML dokumentu.

Deklarace typu dokumentu může být provedena následovně:

- Lokální deklarace:

```
<?xml version="1.0" encoding="windows-1250"?>
<!DOCTYPE type [
<!ELEMENT type (#PCDATA)>
]>
<type>book</type>
```

- externí deklarace³;

```
<?xml version="1.0" encoding="windows-1250"?>
<!DOCTYPE type SYSTEM "type.dtd">
```

- případně jako kombinace výše zmiňovaných možností.

Z ukázky pro externí deklaraci je zřetelné, že musí být pomocí atributu SYSTEM uveden název DTD souboru, ve kterém je definice uvedena. Tento atribut zajistí připojení DTD souboru ke zdrojovému XML. V definici typu dokumentu je možné nalézt následující deklarace:

³Pokud je v deklaraci XML souboru nastaven následující atribut takto: standalone="yes", je využití externí deklarace zakázáno.

- Typu elementu: Tato deklarace elementu zavádí jeho jméno a také specifikuje jeho obsah.

```
<!ELEMENT název_elementu (specifikace obsahu)>
```

Specifikace obsahu elementu se vyjadřuje následovně:

- EMPTY–říká, že daný element je prázdný;
- ANY–říká, že element může mít libovolný obsah;
- (#PCDATA)–tento zápis říká, že element obsahuje řetězec znaků;
- další možností je využití takzvaných operátorů četnosti, které jsou uvedeny v tabulce 1.1.

Tab. 1.1: Vyjádření obsahu elementy operátory četnosti

Operátor četnosti	Význam
'*'	říká, že složka se může libovolně opakovat, včetně nulové iterace
'+'	zaručuje alespoň jedno opakování
'?'	říká, že složka je nepovinná
','	tento operátor nahrazuje symbol konjunkce, nebo-li slovo a, a zároveň pořadí jednotlivých složek je určující
' '	tento operátor nahrazuje symbol disjunkce–slovo nebo

Ukázka zápisu deklarace elementu:

```
<!ELEMENT unit ( type | ... | year | isbn )*>
```

Tento zápis říká, že element unit může ale nemusí obsahovat vypsané elementy a navíc, že jednotka unit se může libovolněkrát ve zdrojovém XML dokumentu opakovat⁴.

- atributů: Deklarace atributů elementů se zavádí následovně:

```
<!ATTLIST název_elementu název_atributu typ_atributu  
volitelnost "implicitní_hodnota">
```

Pro identifikaci elementu slouží následující možnosti:

- ID–určuje unikátnost daného atributu v celém XML dokumentu;

⁴Toto by byl regulerní zápis souboru DTD pro zdrojový text na straně 15, tři tečky jsou pouze pro názornost neomezeného množství elementů.

- CDATA–určuje takzvané prosté řetězce;
- IDREF–slouží k odkazu na existující atribut konkrétního elementu;
- IDREFS–reprezentuje vícehodnotový atribut, v němž jsou jednotlivé hodnoty odděleny bílými znaky, Všechny jeho hodnoty musí být odkazovány na existující elementy dokumentu.

Pro určení volitelnosti slouží následující možnosti:

- REQUIRED–určuje povinnost uvedení u příslušného elementu;
 - #FIXED–určuje, že implicitní hodnota je jediná, které může příslušný atribut nabývat;
 - IMPLIED–určuje nepovinnost uvedení u příslušného elementu.
- entit:⁵ Pokud jsou vytvářené DTD soubory rozsáhlé a složité, je možné využít takzvaných entit a ty se dělí na:
 - Parametrické entity–jsou využitelné pouze v extreních definicích typu dokumentu. Pokud se některý text v souboru DTD opakuje, nabízí se využití parametrických entit;
 - textové entity–slouží pro opakující se části ve zdrojovém XML dokumentu;
 - binární entity–specifické formáty jako jsou obrázky, videa nebo zvuky musí být vkládány pomocí těchto entit.

Jak již bylo řečeno, DTD není jedinou možností pro definici typu dokumentu. V dnešní době je v praxi více využívána možnost využití XSD XML Schema Definition. Toto je zapříčiněno rostoucí rozsáhlostí a komplikovaností potřebných DTD dokumentů. Mezi výhody XSD patří podpora jednoduchých datových typů, má identickou syntaxi jako XML dokumenty, jedná se v podstatě o XML dokument. DTD dokument má jinou syntaxi a nedá se říci, že by byl identický s XML dokumentem. Pro potřeby implementace standardu bibliografických citací $\text{BIB}\text{T}\text{E}\text{X}$ postačí využití DTD dokumentu.

1.2.3 Jmenné prostory

V mnoha zdrojích ke XML technologiím se opomíjí jmenné prostory, a proto je zde o nich základní zmínka. Jak vyplývá z českého ekvivalentu zkratky XML (eXtensible Markup Language), jedná se o rozšiřitelný značkovací jazyk. Při tvorbě XML

⁵V tomto odstavci textu jsou uvedeny pouze základní informace, pokud by čtenáře zajímala tato záležitost do větší holoubky doporučuji pročíst zdroje [7] a [6]

dokumentů neexistuje předdefinovaná sada značek a mohlo by se zdát, že tvůrce má volné pole působnosti při vytváření dokumentu. I když mohou být názvy elementů libovolně pojmenovány (musí být ovšem pamatováno na pravidla pro tvorbu XML dokumentů), nezajistí jedinečnost názvu elementů a atributů tak, aby při zpracování různých XML dokumentů nedocházelo ke kolizím. Ukázky použití jmeného prostoru:

```
xmlns:předpona="URI adresa"  
xmlns:xsl="http://www.w3.org/1999/XSL/Transform"  
xmlns:saxon="http://icl.com/saxon"
```

Pokud je při specifikaci XML dokumentu využit jazyk DTD, musí být jména v rámci této specifikace jedinečná. Stejný název nemohou mít dva elementy, které se liší sadou atributů. Pak nastává problém při kombinaci schémat. Již vyřčená kolize, které se řeší využitím jmenných prostorů, které se uvozují zkratkou xmlns (XML namespace).⁶

1.2.4 Jazyk XPath

Jedná se o takzvaný dotazovací jazyk, který je implementován do dalších standardů XML. Pomocí speciálních dotazů tento jazyk umožňuje vybírat uzly, elementy, atributy, jejich obsahy. Dá se říci, že umožňuje pohyb (operace) ve stromové struktuře XML dokumentu. v této kapitole bude vysvětleno jak jazyk XPath funguje. V převážné části literatury se v případě tohoto jazyka jedná o mnohastránkové popisy [7][6]. Zde jsou teoretické základy z hlediska rozsahu práce shrnuty s co největším ohledem na efektivnost podané informace. Jazyk XPath je velmi důležitý pro práci s dokumenty XSLT. Jeho notace není založena na notaci jazyka XML. Jazyk XPath je základem pro jazyk XPointer a XQuery a již zmiňovaný jazyk XSLT, kterému bude věnována kapitola. Způsob jakým se popisují cesty k elementům, je velmi podobný způsobu popisu cesty operačního systému k souborům uloženým na diskových jednotkách.

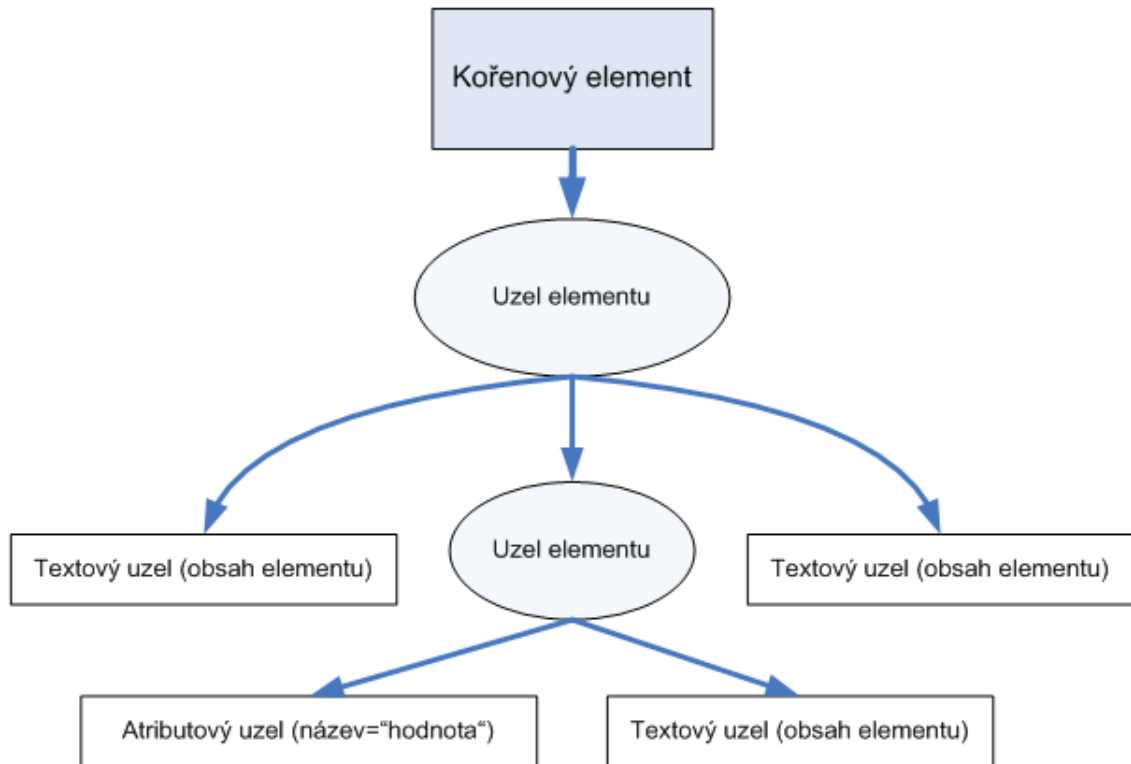
Ve stromově orientované struktuře existují následující typy uzlů:

- Uzel označující kořen dokumentu;
- uzel elementu;
- textový uzel;
- uzel atributu;

⁶Podrobnější informace o jmenných prostorech lze nalézt v literatuře [6]. Je zde pojednáno o explicitních i implicitních jmenných prostorech spolu s uvedením několika příkladů a problematika je poměrně podrobně rozebrána.

- uzel komentáře;
- uzel instrukce pro zpracování;
- uzel jmenného prostoru.

Ukázka stromové struktury a některých uzlů je uvedena na obrázku 1.1



Obr. 1.1: Jazyk XPath–ukázka XML stromu spolu s uzly

Základní typy XPath dotazů jsou uvedeny v tabulce 1.2

Dle literatury [7] pro oslovení atributů jazykem XPath se používá znaku '@'. Využije-li se některého z fundamentálních zápisů dotazů XPath uvedených v tabulce 1.2, může zápis vypadat následovně:

`//@*`

Tento zápis umožňuje oslovit libovolný atribut na jakékoliv úrovni dokumentu. Pokud by v dokumentu například existoval atribut s názvem `sequence` a byl místo znaku `*` uveden například řetězec `'seq'`, budou vybrány veškeré atributy elementů dokumentu, které začínají písmeny `seq`.

Pro výběr obsahu elementu slouží následující zápis:

Tab. 1.2: Příklady XPath dotazů

Příklad	Vysvětlení
<code>/bibliography/unit</code>	uvozuje absolutní cestu, která vrací uzel odpovídající elementu unit na nejvyšší úrovni v dokumentu
<code>unit/isbn</code>	uvozuje relativní cestu, jejímž výsledkem jsou všechny elementy isbn, které jsou dítětem běžného uzlu
<code>/bibliography/unit//type</code>	uvozuje relativní cestu, jejímž výsledkem jsou všechny elementy type, které jsou potomky elementu unit
<code>/unit</code>	vybere každý element unit, vyskytující se v libovolné úrovni dokumentu
<code>/*/*/type</code>	budou vybrány všechny elementy třetí úrovně s názvem type. Pokud by například na čtvrté úrovni existovaly elementy type, jejich existence by nebyla brána v potaz a vybrala by se pouze třetí úroveň

`/bibliography/unit/isbn/text()`

Tato varianta zápisu vybere obsahy elementů isbn, tedy všechna čísla ISBN která jsou uvedena v XML dokumentu,

Pomocí jazyka XPath lze také vytvářet podmínky, které se do zapisovaného výrazu vkládají do hranatých závorek. Podmínka výběru lze v jednom zápisu využít i vícekrát. Výběrové podmínky je možné samozřejmě kombinovat s atributy a jejich hodnotami. Více informací v literatuře [7] a [6], stejně tak jako informace o využitelných funkcích ve výrazech XPath.

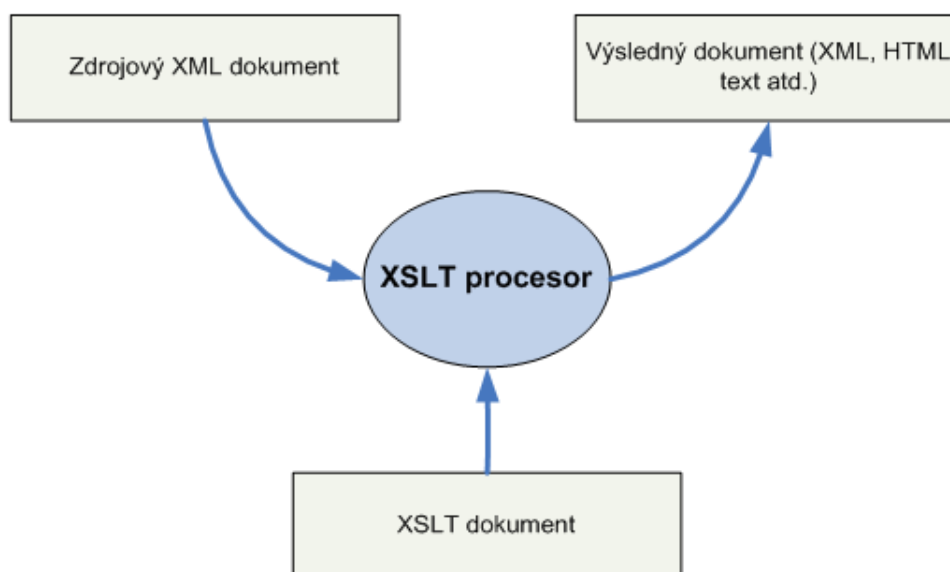
1.2.5 XSL

Jednou z velmi důležitých částí XML technologie je XSL (eXtensible Stylesheet Language), je to stylový prostředek, který využívá syntaxe jazyka XML. Dle konsorcia W3C se jazyk XSL skládá ze dvou částí:

- XSLT–eXtensible Stylesheet Language Transformations, českým ekvivalentem metody pro transformaci XML dokumentů;
- XSLFO–eXtensible Stylesheet Language Formatting Objects, českým ekvivalentem metody formátování XML dokumentů.

Dále se tento text bude zabývat pouze XSLT dokumenty. Zásadní fakt je, že XML dokument obsahuje zdrojová data a ta se dají získávat pomocí zmiňovaných XSLT souborů, které prakticky neobsahují data až na výjimku elementu `xsl:text`.

XSLT je v současné době nejdůležitější a nejvíce využívanou částí standardu XSL. Umožňuje provést transformaci XML dokumentu na jiný XML dokument. Pomocí jazyka XSLT mohou být odstraňovány, vytvářeny, používány, přeskupovány a tříděny prvky ze zdrojového XML dokumentu. Při transformačním procesu jsou používány výrazy XPath k získání informací ze zdrojového dokumentu. Při tvorbě XSLT dokumentů musí být brány v potaz všechny podmínky pro tvorbu korektních dokumentů XML (konkrétněji dále v textu). Transformace pomocí XSLT je zobrazena na obrázku 1.2.



Obr. 1.2: Transformace pomocí XSLT

Je vhodné definovat XSLT procesor. Je to prostředek, který aplikuje XSLT dokument na zdrojové XML data. Může jím být například webový prohlížeč Internet Explorer od verze 5.0, která má ovšem určitá omezení. Dále lze využívat takzvané parsery.

Použití jazyka XSLT musí splňovat následující pravidla[7]:

- Každý XSLT dokument by měl začínat deklarací, minimálně by měl obsahovat deklaraci verze a podle užitých znaků i kódování, zápis je identický jako pro zdrojové XML;
- dokument XSLT musí obsahovat právě jeden kořenový element s deklarací jmenného prostoru;

Příklad kořenového elementu XSLT⁷ :

⁷Narozdíl od deklarace verze pro XML dokument, nemusí být deklarace verze na prním místě jako v případě popsaném v kapitole 1.2, zapříčiněno je to tím, že se v podstatě jedná o atributy.

```
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" ...
version="1.0">
</xsl:stylesheet>
```

- všechny elementy v XSLT dokumentu musejí být párové, případně prázdné se správným syntaktickým zápisem.

Pojednat o problematice XSLT souborů by mohlo vydat na mnohastránkové dokumenty, proto dále v textu budou vypsány pouze základní elementy pro použití v XSLT dokumentech. Hlavním zdrojem inspirace byla literatura [7] a [6], pokud by bylo potřeba nalézt podrobnější informace, doporučuji www.w3schools.com:

- **xsl:stylesheet**—jedná se o kořenový element, který má povinný atribut `version`. Názorná ukázka použití je uvedena v příkladu kořenového elementu, ke kterému je také vedena poznámka číslo 9. Tento zápis lze nahradit zápisem `xsl:transform`. Jedná se pouze o jiný zápis kořenového elementu. I u tohoto zápisu se deklaruje verze a jmenné prostory. Pokud se v tomto zápisu udělá chyba, nedojde k požadovanému transformačnímu procesu. Je velmi důležité si uvědomit, že jazyk XSLT je case-sensitive, tedy rozlišuje malá a velká písmena;
- **xsl:output**—tento element ovlivňuje výstupní formát XML dokumentu, který musí být umístěn hned za počátečním tagem elementu `xsl:stylesheet`. Formát výstupního dokumentu lze řídit několika atributy. Základní atributy jsou uvedeny v tabulce 1.3, podkladem pro její vytvoření byly zdroje;

Ukázka zápisu elementu `xsl:output`:

```
<xsl:output method="text" encoding="windows-1250"...
omit-xml-declaration="yes" indent="no"/>
```

- **xsl:template**—tento element je základním stavebním kamenem XSLT dokumentů. Tento element má povinný atribut s názvem `match`, pokud není uveden tento atribut je nutné jej nahradit jiným povinným atributem. Náhradou může být atribut `name`, který umožňuje pojmenovat šablonu pro využití v jiných XSLT elementech. Základem atributu `match` je v uvozovkách uvedená potřebná cesta XPath XML dokumentem;
- **xsl:apply-templates**—z předchozího textu je známo, že XML technologie mají stromovou strukturu. Pomocí tohoto elementu je možné projít celou strukturu až do nejnižší možné úrovně. Než tedy přejde šablona na další sousední element (sourozenecký) projde předešlý element až do nejnižší úrovně. Tento element nemá počáteční a konečný tag a zapisuje se jako prázdný element; *Ukázka zápisu elementu `xsl:apply-templates`:*

Tab. 1.3: XSLT - Vybrané atributy elementu xsl:output

Atribut	Hodnoty	Význam
method	"html", "xml", "text"	jedná se o základní atribut pro definici výstupního dokumentu. Hodnoty atributu vypovídají jednoznačně o tom, v jakém formátu bude výstup generován
version	např. "1.0"	pomocí tohoto atributu lze určit číslo verze výstupního formátu
encoding	"windows-1250", "UTF-8", "iso-8859-2" atd.	určuje požadované kódování výstupního formátu
indent	"yes", "no"	určuje zda bude ve zdrojovém kódu výstupního dokumentu zapsán každý element na novém řádku
omit-xml-declaration	"yes", "no"	při vynechání tohoto atributu a použití jiného kódování než UTF-8 a UTF-16 může být výstupní dokument nesprávně strukturovaný

`<xsl:apply-templates/>`

- **xsl:value-of**—tímto elementem se zajišťuje vypsání textového obsahu zdrojového XML dokumentu do výstupního dokumentu. Povinným atributem je u tohoto elementu takzvaný select, ke kterému se dle již známých pravidel zapisuje cesta XPath k požadovanému elementu ve zdrojovém XML dokumentu;
- **xsl:text**—pomocí tohoto elementu se do výstupního dokumentu zapisuje potřebný text, který není uveden ve zdrojovém XML dokumentu;
- **xsl:if**—tento element slouží pro vytvoření podmínky v rámci transformace vstupního dokumentu. Bude-li splněna podmínka vypsána pomocí jazyka XPath, provede se vypsání kódu mezi počátečním a koncovým tagem tohoto elementu. Atribut který určuje podmínku, se nazývá test. Tento element má identickou funkci jako ve většině programovacích jazyků příkaz if. Jediným rozdílem je, že element xsl:if neumí řešit stav else, tedy situaci, kdy daná podmínka neplatí;
- **xsl:for-each**—tento element slouží k vytvoření cyklického opakování příkazů nebo-li takzvané iteraci. Tento element má povinný atribut select. Ukázka použití tohoto elementu je uvedena u elementu xsl:sort;

- **xsl:sort**—tento atribut slouží k setřídění prvků ve výstupním dokumentu. Nemá žádný povinný atribut, ale má 5 nepovinných atributů. Pokud se neuvede žádný atribut, vykoná element `xsl:sort` třídění. Selektce bude provedena dle aktuálního prvku vzestupně dle abecedy. Tento element se musí využívat jako vnořený v těle elementů `xsl:apply-templates`, nebo `xsl:for-each`. Bližší informace o nepovinných elementech jsou uvedeny v tabulce 1.4;

Ukázka zápisu elementu `xsl:sort` bez nepovinných atributů, tedy vzestupné abecední třídění dle aktuálního prvku:

```
<xsl:for-each select="XPath výraz">
  <xsl:sort/>
  .
  .
  .
</xsl:for-each>
```

Tab. 1.4: Atributy elementu `xsl:sort`

Atribut	Popis
<code>select</code>	umožňuje bližší specifikaci klíče pro třídění dle XPath výrazů
<code>order</code>	dle svých hodnot definuje sestupné (descending) či vzestupné (ascending) řazení prvků
<code>data-type</code>	určuje dle jakého typu dat bude výstup tříděn
<code>lang</code>	identifikuje použitý jazyk a pomocí <code>case-order</code> umožňuje upřednostnit řazení dle malých (lower-first) nebo velkých písmen (upper-first)

1.3 Bib_TE_X

V této kapitole bude čtenáři zběžně představen program $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ a $\text{BibT}_{\text{E}}\text{X}$. V první části bude o těchto programech pojednáno obecně a v kapitole 1.3.2 bude popsáno jeho využití a způsob inicializace. Je podstatné si uvědomit, že ani jeden ze zmiňovaných programů neumí pracovat s XML soubory, protože programy $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ a $\text{BibT}_{\text{E}}\text{X}$ vznikly dříve než XML technologie. Tyto programy i XML technologie se zakládají na principu značkování.

1.3.1 Ve zkratce o programu \LaTeX a BibTeX

\LaTeX je profesionální sázecí program pro tvorbu matematických a obecně technických publikací. Je to nadstavba programu $\text{T}_{\text{E}}\text{X}$. Jeho vstupem je libovolný textový soubor pořízený například v programu TeXniCenter nebo PSPad či v jiných. Tyto textové soubory mají příponu `tex`. Vedle vlastního textu se do těchto souborů zapisují příkazy, které určují vlastní vysázení textu, tedy výslednou podobu po kompilaci [9], [8].

BibTeX je program, který \LaTeX využívá pro vytváření seznamu dokumentu a citování zdrojů. Oproti ruční tvorbě seznamů použité literatury, která využívá příkazu `\thebibliography`, skýtá program BibTeX řadu výhod. Seznam použité literatury je uveden v jednom souboru. Do seznamu použité literatury program BibTeX vypisuje pouze ty zdroje, které byly v dokumentu využity.

1.3.2 Aplikace programu BibTeX

BibTeX ová databáze je do programu \LaTeX ového zdrojového kódu vkládána pomocí příkazu `\bibliography` ve stejném místě, kde by byl normálně použit příkaz `\thebibliography` a to tímto způsobem:

```
\bibliographystyle{styl}  
\bibliography{jméno databáze}
```

styl je název požadovaného stylu formátování referencí [5]. Základní styly jsou uvedeny v tabulce 1.5: *jméno databáze* je název BibTeX ového souboru s příponou

Tab. 1.5: Styly pro formátování referencí programu BibTeX

Styl	Popis
alpha	citace jsou abecedně seříděny, ale označují se zkratkou vytvořenou z prvních písmen příjmení prvních tří autorů a z roku vydání
plain	citace abecedně seřídí a označuje je číslem
unsrt	uvádí citace v pořadí, v jakém jsou citovány v textu a citace označuje číslem
abbrv	tento styl pracuje stejně jako styl alpha, ale místo plných jmen autorů uvádí jejich první písmena
czechiso	tento styl je počestěn a odpovídá normě ČSN ISO 690
csplainnant	tento styl je počestěn a odpovídá normě ČSN ISO 690

`.bib`, která se musí nacházet ve stejném adresáři jako zdrojový soubor \LaTeX ového

dokumentu. Tento soubor obsahuje strukturované údaje, které mohou vypadat následovně:

```
@BOOK{xml_upl_zac,  
  title = {XML pro úplné začátečníky},  
  publisher = {Computer Press},  
  year = {2002},  
  author = {Lucie Grusová},  
  address = {Praha},  
  edition = {Vydání první},  
  isbn = {80-7226-697-7}  
}
```

```
@BOOK{latex_pro_zac,  
  title = {Latex pro začátečníky},  
  publisher = {Konvoj},  
  year = {2003},  
  author = {Jiří Rybička},  
  address = {Brno},  
  edition = {Třetí vydání},  
  isbn = {80-7302-049-1}  
}
```

Zaznamenávají se informace jako například jméno autora, typ citace, rok vydání, atd. V zápisu @BOOK se zavináčem uvozuje typ citace. Typů záznamu je celá řada, a to například: (@BOOK, @MANUAL, @ARTICLE, @INPROCEEDINGS, @ELECTRONIC, @MANUAL). Každý typ má odlišné položky a také formát v jakém bude vysázen. Dále následuje návěstí (xml_upl_zac, latex_pro_zac), pomocí kterého se na zdroj bude odkazovat s využitím příkazu \cite{}. Zápis může tedy vypadat takto:

```
\cite{latex_pro_zac}
```

Dále jsou uvedeny jednotlivé položky (informace o zdroji), které jsou uvedeny ve složených závorkách nebo mohou být uvozeny dvojitými uvozovkami. Jednotlivé položky musí být ukončeny čárkou. Vytvářet zdrojový soubor je možné v jakémkoliv textovém editoru, ale jsou i programy, které jsou na tuto činnost specializované. Pro tvorbu BIB_TE_Xové databáze slouží například program JabRef[5].

Při generování výsledného výstupu je třeba provádět překlady víceprůchodově. V prvním průchodu vygeneruje L^AT_EX požadavky na odkazy, které se pak zpracují pomocí BIB_TE_Xu a výsledek je potřeba podruhé zkompileovat L^AT_EXem. Například

mějme v adresáři soubory `sablona.tex` a `used_bibliography.bib`, překlad pak bude vypadat následovně:

```
pdfcslatex sablona
bibtex sablona
pdfcslatex sablona
pdfcslatex sablona
```

Po provedení příkazu `bibtex` se vygeneruje do stejného adresáře, ve kterém jsou uložena zdrojová data soubor s příponou `.bib`, v tomto případě `sablona.bib`. Obsah tohoto souboru pak \LaTeX při překladu použije v místě příkazu `\bibliography`.

2 IMPLEMENTACE CITACÍ POMOCÍ XML TECHNOLOGIÍ

V této části textu je popsána vlastní tvorba souboru pro $\text{BIB}\text{T}_{\text{E}}\text{X}$ s využitím XML technologií a také vygenerování seznamu citovaných zdrojů ze zdrojového XML dokumentu. Nejprve je popsán proces, kterým jsou výsledné výstupní soubory získány. Následně jsou rozebrány jednotlivé soubory které se v implementaci využívají (DTD, XML, XSL). Jsou definovány základní části jak z pohledu typů citovaných zdrojů, tak z pohledu samotné syntaxe dokumentů. Dále jsou také uvedeny ukázky výstupních souborů (bib, tex, html). V poslední části je uvedena metodika pro práci s vytvořeným programem, který umožňuje grafické ovládání parseru Saxon.

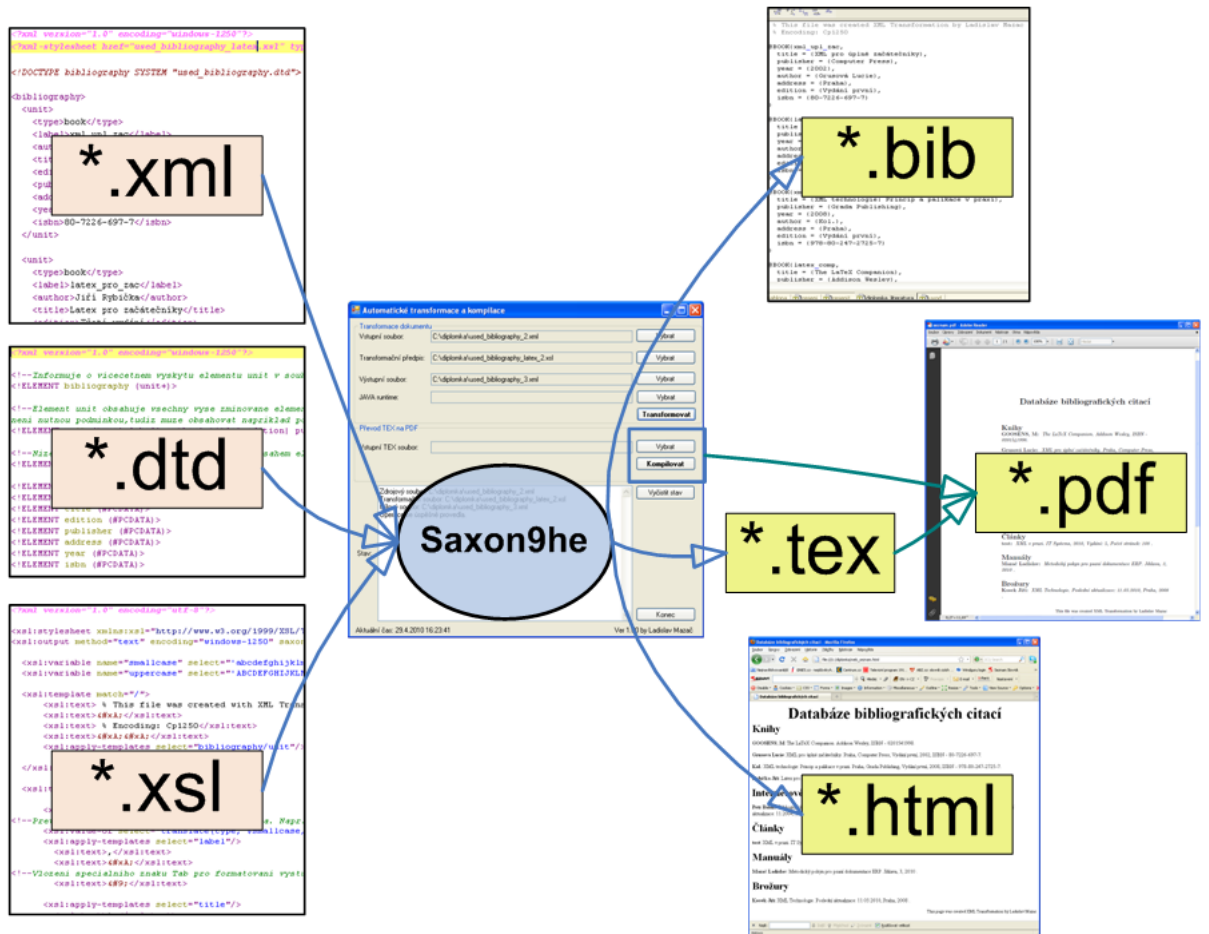
2.1 Proces implementace

Proces samotné tvorby $\text{BIB}\text{T}_{\text{E}}\text{X}$ ového souboru je uveden na obrázku 2.1.

Jako XSLT procesor je použit parser Saxon9he, který zajistí vytvoření souboru s požadovanou příponou. Aplikuje na zdrojový dokument *.xml šablonu s názvem *.xsl. Dále je na zdrojový soubor aplikován soubor *.dtd, který zajišťuje validitu zdrojového XML dokumentu. Parser Saxon a tedy samotná tvorba výstupních souborů se provede zapsáním následující syntaxe do příkazového řádku, případně lze využít vytvořený grafický program pro ovládání parseru, který je popsán v kapitole 2.4. Názvy souborů jsou pouze ukázkové.

```
java -cp . -jar saxon9he.jar -expand:on -o:used_bibliography.bib  
used_bibliography.xml used_bibliography_latex_.xsl
```

Výstupními soubory tedy jsou *.bib, *.tex, *.html jak je viditelné na obrázku 2.1. Soubory *.bib jsou plně funkční a použitelné pro citace zdrojů v programu $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$.



Obr. 2.1: Proces implementace citací s využitím XML technologií a grafického rozhraní

2.2 Zdrojové soubory

Typy citací, které lze do databáze bibliografických citací vytvořené pomocí XML technologií v této diplomové práci zahrnout, jsou následující:

- Kniha;
- článek;
- internetový zdroj;
- manuál;
- brožura;
- standard.

Pokud je některý typ z těchto citací uveden ve zdrojovém XML dokumentu a je na něj aplikována XSL šablona, vygeneruje se konkrétní zdroj citace na požadovaném výstupu. Zároveň jsou typy bibliografických citací deklarovány v DTD dokumentu. Pro načerpání dostatečného know-how a řešení problémů při tvorbě zdrojových kódů byly použity následující zdroje: [7], [6], [11, 1, 13, 14, 4, 12, 10].

V následujících kapitolách jsou uvedeny a popsány základní pilíře jednotlivých souborů. Jako důkaz funkčnosti výstupního *.bib souboru je v této diplomové práci použita vygenerovaná databáze bibliografických citací ze zdrojového XML dokumentu.

2.2.1 DTD dokument

Document Type Definition říká, jak by měla vypadat podoba zdrojového XML dokumentu. DTD dokument má tři základní části. První částí je definice obsahu kořenového elementu. Je tedy řečeno, že kořenový element s názvem `bibliography` bude obsahovat elementy (typy citací) `book`, `article`, `electronic`, `manual`, `booklet`, `standard`. Dále je definováno, že elementy se mohou vyskytovat vícekrát, případně se nemusí vyskytovat vůbec. Zápis je následující:

```
<!ELEMENT bibliography ( (book | ... |standard)*)
```

Druhou částí je definice atributů jednotlivých citací. Je tedy řečeno které elementy budou obsahovat jednotlivé typy bibliografických citací. Je pevně dáno pořadí elementů ve zdrojovém XML dokumentu, to zajišťuje oddělení elementů čárkami. Otazník za jednotlivými elementy (atributy) říká, že atribut je nepovinný. V rámci této

definice je ponechán jako povinný pouze element label, což je zkratka, která se používá pro uvedení citace v programu L^AT_EX. Je uvedena pouze ukázka deklarace typu citace kniha (book)⁸:

```
<!ELEMENT book (label, (author? , title? ,..., isbn?))>
```

Naprosty identicky jsou uvedeny definice všech ostatních typů citací (article, electronic, manual, booklet, standard).

Třetí a poslední důležitou částí je definice obsahu jednotlivých elementů (atributů), která se provede následovně:

```
<!ELEMENT label (#PCDATA)>
<!ELEMENT author (#PCDATA)>
<!ELEMENT title (#PCDATA)>
...
...
...
<!ELEMENT organization (#PCDATA)>
<!ELEMENT month (#PCDATA)>
<!ELEMENT institution (#PCDATA)>
```

V poslední části musí být definovány přímé obsahy všech použitých elementů (atributů), které jsou uvedeny v kroku dvě.

2.2.2 XML dokument

Citované zdroje jsou uloženy v souboru s příponou xml. Z tohoto souboru je také lépe viditelné, jaké elementy (atributy) jednotlivé typy citací obsahují⁹.

- **Kniha:**

```
<book>
  <label>latex_pro_zac</label>
  <author>Jiří Rybička</author>
  <title>Latex pro začátečníky</title>
  <edition>Třetí vydání</edition>
  <publisher>Konvoj</publisher>
  <address>Brno</address>
  <year>2003</year>
```

⁸Konkrétních elementy (atributy) bibliografické citace kniha (book) a všech ostatních typů citací, je blíže rozepsána v kapitole 2.2.2

⁹Samotná data obsažená v elementech (atributech) jednotlivých citací mohou být smyšlená

```
<isbn>80-7302-049-1</isbn>
</book>
```

- **článek:**

```
<article>
  <label>it_systems</label>
  <author>test</author>
  <title>XML v praxi</title>
  <journal>IT Systems</journal>
  <year>2010</year>
  <volume>5</volume>
  <pages>108</pages>
</article>
```

- **internetový zdroj:**

```
<electronic>
  <label>w3_schoo</label>
  <author></author>
  <title>W3Schools Online Web Tutorials</title>
  <version></version>
  <updating></updating>
  <url>http://w3schools.com/</url>
  <cite>[cit. 2009-10-13]</cite>
</electronic>
```

- **manuál:**

```
<manual>
  <label>man_test</label>
  <author>Mazač Ladislav</author>
  <title>Metodický pokyn pro psaní dokumentace ERP</title>
  <organization></organization>
  <address>Jihlava</address>
  <edition>3</edition>
  <month>4</month>
  <year>2010</year>
</manual>
```

- **brožura:**

```
<booklet>
  <label>b_test</label>
  <author>Kosek Jiří</author>
  <title>XML Technologie</title>
  <updating>11.03.2010</updating>
  <address>Praha</address>
  <year>2008</year>
</booklet>
```

- **standard:**

```
<standard>
  <label>itil_s</label>
  <author>ITIL company</author>
  <title>ITIL foundation</title>
  <organization></organization>
  <institution></institution>
  <year>2009</year>
</standard>
```

Pokud by bylo za potřeby dalších atributů u jakéhokoliv typu citace, musel by se doplnit do DTD souboru a také do zdrojového XML dokumentu. Další nutná úprava by spočívala v rozšíření XSLT šablon pro transformaci.

2.2.3 XSL dokument pro výstupní soubor bib

Zvláštností této šablony je to, že připravuje výstup pro další značkovací jazyk. BIB_{TEX} také pracuje na bázi značkovacích jazyků jak je uvedeno v kapitole 1.3. Je tedy zapotřebí pomocí XSL souboru vhodně transformovat data ze zdrojového XML souboru do výstupního souboru `bib`. V rámci této práce budou uvedeny ukázky zdrojových kódů pouze na jeden typ citace a to na typ kniha (book). Řešení ostatních typů citací je identické. K diplomové práci jsou kompletní zdrojové kódy přiloženy. Je nutné zdůraznit, že pro aktivní funkci internetových odkazů uvedených v použité literatuře této práce, není odkaz URL uveden v závorkách `<>`. V ostatních výstupech `tex`, `pdf`, `html` jsou dle norem závorky použity.

Tento transformační soubor je hlavní částí této diplomové práce. Je z vytvořených XSL transformačních souborů také nejsložitější. Jeho struktura má několik základních částí, které budou dále jmenovány a popsány. Jsou zde uvedeny zdrojové kódy, které jsou slovně okomentovány. Celková ukázka části zdrojového kódu je uvedena na konci této kapitoly.

- **Atributy elementu `xsl:stylesheet`:**

Tab. 2.1: Atributy elementu `xsl:stylesheet` pro výstupní *.bib soubor

Atribut	Hodnota	Popis
<code>xmlns:xsl</code>	"http://www.w3.org/1999/XSL/Transform"	deklarace transformačního souboru
<code>version</code>	"2.0"	určení verze XML technologie
<code>xmlns:saxon</code>	"http://icl.com/saxon"	potřebná deklarace parseru saxon
<code>extension-element-prefixes</code>	"saxon"	potřebná deklarace parseru saxon

- **Atributy elementu `xsl:output`:**

Tab. 2.2: Atributy elementu `xsl:output` pro výstupní *.bib soubor

Atribut	Hodnota	Popis
<code>method</code>	"text"	základní atribut definující textový výstup
<code>encoding</code>	"windows-1250"	určení typu kodování XML dokumentu
<code>saxon:character-representation</code>	"native"	deklarace pro změnu výstupního kódování
<code>indent</code>	"yes"	zajištění formátování výstupu
<code>omit-xml-declaration</code>	"yes"	zajišťuje správnou strukturovanost při využití kódování typu UTF

- **Zajištění převodu typu bibliografických citací na velká písmena:**

```
<xsl:variable name="smallcase" select="'abcdefghijklmnopqrstuvwxyz' " />
<xsl:variable name="uppercase" select="'ABCDEFGHIJKLMNOPQRSTUVWXYZ' " />
...
...
...
<xsl:value-of select="translate(name(.), $smallcase, $uppercase)" />
```

Nejprve musí být deklarovány proměné pro malá a velká písmena a následně je v elementu `xsl:value-of` použit převod na velká tiskací písmena. Názvy elementů (typů bibliografických citací) jsou vygenerovány do výstupního souboru velkým tiskacím písmem stejně tak, jako by byl zdroj vložen do databáze bibliografických citací pomocí programu JabRef.

- **Využití elementu `xsl:template` a `xsl:apply-template`:**

Tato část je úplně základním stavebním kamenem této transformace. Konkrétní ukázka je uvedena na konci této kapitoly.

- **Využití elementu `xsl:if`:**

```
<xsl:if test="normalize-space(title)">
  <xsl:apply-templates select="title"/>
  <xsl:text>},</xsl:text>
  <xsl:text>&#xA;</xsl:text>
  <xsl:text>&#9;</xsl:text>
</xsl:if>
...
...
...
<xsl:template match="title">
  <xsl:text>title&#32;=&#32;{</xsl:text>
  <xsl:value-of select="title"/>
  <xsl:apply-templates/>
</xsl:template>
```

Výše uvedený zápis říká, že XSL transformace uvedená mezi počátečním a koncovým elementem `xsl:if` se provede pouze tehdy, bude-li element `title` obsahovat

jakýkoliv znak. Pokud by nebyl tento element použit, převedla by se do výstupního souboru deklarace pro BIBTEXový značkový jazyk zbytečně. Jednotlivé zdroje, uvedené v databázi bibliografických citací, by tedy vypadaly následovně:

```
@BOOK{latex_comp,
  title = {The LaTeX Companion},
  publisher = {Addison Wesley},
  year = {},
  author = {GOOSENS, M},
  address = {},
  edition = {},
  isbn = {0201541998}
}
```

Pokud ve zdrojovém XML dokumentu (uložené zdroje bibliografických citací) neobsahují některé zdroje všechny atributy, tak právě aplikací elementu xsl:if, pak výstupní soubor neuvádí BIBTEXovou deklaraci všech atributů citací a vypadá následovně:

```
@BOOK{latex_comp,
  title = {The LaTeX Companion},
  publisher = {Addison Wesley},
  author = {GOOSENS, M},
  isbn = {0201541998}
}
```

- **Využití číselných entit a znaků k úpravě výstupního bib souboru:**
Využitím číselných entit a speciálních znaků bylo dosaženo formátování jednotlivých zdrojů v databázi bibliografických citací. Bib soubor je díky tomu formátován naprosto identicky jako by byl vytvořen v programu JabRef. K výpisu potřebných textů (znaků) je použit element xsl:text.

Zkrácená ukázka zdrojového kódu na typ bibliografické citace book (kniha) je následující:

```
<xsl:template match="/">
  <xsl:text>% This file was created XML Transfo.....
  <xsl:text>&#xA;</xsl:text>
  <xsl:text> % Encoding: Cp1250</xsl:text>
  <xsl:text>&#xA;&#xA;</xsl:text>
```

Tab. 2.3: Číselné entity, formátovací znaky a jejich popis

Číselná hodnota, znak	Popis
<code>&#xA;</code>	vkládá do výstupního souboru speciální znak "Enter"
<code>&#9;</code>	vkládá do výstupního souboru speciální znak "Tab"
<code>&#32;</code>	vkládá do výstupního souboru speciální znak "white space", mezeru
<code>@</code>	deklaruje počátek zdroje (citace) v souboru bib
<code>,</code>	odděluje jednotlivé atributy (položky) jednotlivých zdrojů v databázi bibliografických citací
<code>{}</code>	ve složených závorkách jsou uvedeny konkrétní údaje, které jsou pak vysázené v programu L ^A T _E X

```

<xsl:apply-templates select="bibliography/book"/>
...
...
...
<xsl:apply-templates select="bibliography/standard"/>
</xsl:template>

<xsl:template match="book">
  <xsl:text>@</xsl:text>
  <xsl:value-of select="translate(name(.), $smallcase, $uppercase)" />
  <xsl:apply-templates select="label"/>
  <xsl:text>,</xsl:text>
  <xsl:text>&#xA;</xsl:text>
  <xsl:text>&#9;</xsl:text>
  <xsl:if test="normalize-space(title)">
    <xsl:apply-templates select="title"/>
    <xsl:text>},</xsl:text>
  <xsl:text>&#xA;</xsl:text>
  <xsl:text>&#9;</xsl:text>
</xsl:if>
  <xsl:if test="normalize-space(publisher)">
    <xsl:apply-templates select="publisher"/>
    <xsl:text>},</xsl:text>
  <xsl:text>&#xA;</xsl:text>
  <xsl:text>&#9;</xsl:text>

```

```

    </xsl:if>
    ...
    ...
    ...
    <xsl:if test="normalize-space(isbn)">
      <xsl:apply-templates select="isbn"/>
      <xsl:text>}</xsl:text>
      <xsl:text>&#xA;</xsl:text>
    </xsl:if>
    <xsl:text>}</xsl:text>
    <xsl:text>&#xA;</xsl:text>
    <xsl:text>&#xA;</xsl:text>
  </xsl:template>

  <xsl:template match="label">
    <xsl:text>{</xsl:text>
    <xsl:value-of select="label"/>
    <xsl:apply-templates/>
  </xsl:template>

  <xsl:template match="title">
    <xsl:text>title&#32;=&#32;{</xsl:text>
    <xsl:value-of select="title"/>
    <xsl:apply-templates/>
  </xsl:template>

  <xsl:template match="publisher">
    <xsl:text>publisher&#32;=&#32;{</xsl:text>
    <xsl:value-of select="publisher"/>
    <xsl:apply-templates/>
  </xsl:template>

  ...
  ...
  ...
  <xsl:template match="isbn">
    <xsl:text>isbn&#32;=&#32;{</xsl:text>
    <xsl:value-of select="isbn"/>
    <xsl:apply-templates/>
  </xsl:template>

```


2.2.4 XSL dokument pro výstupní soubor tex

Další vytvořenou XSL šablonou je transformační předpis pro vytvoření výstupního tex souboru, ze kterého lze následně vytvořit výstupy, které mohou být generovány za pomoci sázecího programu \LaTeX (* .pdf, * .dvi atd.). Pomocí tohoto XSL souboru lze přehledně vygenerovat seznam citovaných zdrojů.

Tato šablona má také své specifické a důležité části, které jsou zde shrnuty:

- **Atributy elementu `xsl:stylesheet`:**

Tab. 2.4: Atributy elementu `xsl:stylesheet` pro výstupní *.tex soubor

Atribut	Hodnota	Popis
<code>xmlns:xsl</code>	"http://www.w3.org/1999/XSL/Transform"	deklarace transformačního souboru
<code>version</code>	"2.0"	určení verze XML technologie

- **Atributy elementu `xsl:output`:**

Tab. 2.5: Atributy elementu `xsl:output` pro výstupní *.tex soubor

Atribut	Hodnota	Popis
<code>method</code>	"text"	základní atribut definující textový výstup
<code>encoding</code>	"windows-1250"	určení typu kodování XML dokumentu
<code>omit-xml-declaration</code>	"yes"	zajišťuje správnou strukturovanost při využití kódování typu UTF

- **Využití elementu `xsl:template` a `xsl:apply-template`:**

Tato část je rovněž jako u XSL šablony pro *.bib soubory základním stavebním kamenem transformace. Konkrétní ukázka je uvedena na konci této kapitoly.

- **Využití příkazů z programu \LaTeX :**

Tyto příkazy slouží pro deklaraci rozměru \LaTeX ového výstupu a pro jeho použité kódování.

```

\documentclass[a4paper,pdftex]{article}
\usepackage[Cp1250]{inputenc}
\usepackage{czech}

```

Dále jsou použity následující příkazy: K výpisu potřebných textů (znaků) je

Tab. 2.6: Použité příkazy sázecího programu L^AT_EX

Příkaz	Popis
<code>\[10mm]</code>	vloží do textu vertikální mezeru dle uvedených hodnot v hranatých závorkách
<code>\noindent</code>	zahájí odstavec bez odstavcové zarážky
<code>center{}</code>	zarovná vypsaný text ve složených závorkách na střed
<code>flushleft{}</code>	zarovná vypsaný text ve složených závorkách na levo
<code>textbf{}</code>	text uvedený ve složených závorkách je vysázen tučně
<code>Large{}</code>	text uvedený ve složených závorkách je vysázen velikostí 14,4 pt (velikost písma v textových editorech)
<code>LARGE{}</code>	text uvedený ve složených závorkách je vysázen velikostí 17,28 pt

použit element `xsl:text`.

- **Využití elementu `xsl:sort`:**

```
<xsl:sort select="author"/>
```

Tento element seřadí abecedně vzestupně zdroje bibliografických citací v databázi podle elementu `author`.

- **Využití elementu `xsl:if`:**

U této šablony se element `xsl:if` využívá proto, aby nebyl nadpis typu citace (např. Kniha) vysázen, pokud zdrojový XML soubor neobsahuje element (typ citace) `book`. Dále je použit pro formátování. Jednotlivé atributy konkrétní citace mají dle norem ČSN 690 dané pořadí a také oddělovací znaky. Pokud by ve zdrojovém XML souboru neměl některý z elementů obsah (atribut by nebyl uveden), sázely by se na výstup oddělovací znaky. Proto je zde také využito tohoto elementu. Názorné využití tohoto elementu je zřetelné na ukázce zdrojového kódu na konci této kapitoly.

Zkrácená ukázka zdrojového kódu pro generování výstupních *.tex souborů:

```

\begin{document}%
\center{\textbf{\LARGE{Databáze bibliografických citací}}}

```

```

\\[10mm]
\noindent
<xsl:if test="bibliography/book">
\flushleft{\Large{\textbf{Knihy}}}
</xsl:if>
<xsl:apply-templates select="bibliography/book">
<xsl:sort select="author"/>
</xsl:apply-templates>
...
...
...
<xsl:apply-templates select="bibliography/standard">
<xsl:sort select="author"/>
</xsl:apply-templates>
\flushright{\footnotesize
\noindent
This file was created XML Transformation by Ladislav Mazac
}
\end{document}
...
...
...
<xsl:template match="book">
  \textbf{<xsl:if test="normalize-space(author)">
    <xsl:value-of select="author"/>
    <xsl:text>: </xsl:text>
  </xsl:if>}
  \textit{<xsl:if test="normalize-space(title)">
    <xsl:value-of select="title"/>
    <xsl:text>.</xsl:text>
  </xsl:if>}
  ...
  ...
  ...
  <xsl:if test="normalize-space(isbn)">
    ISBN -
    <xsl:value-of select="isbn"/>
    <xsl:text>.</xsl:text>
  </xsl:if>

```

```

    \\[3mm]
    %
</xsl:template>
...
...
...

```

2.2.5 XSL dokument pro výstupní soubor html

Poslední vytvořenou XSL šablonou je transformační předpis pro vytvoření výstupního *.html souboru. Jedná se o seznam zdrojů obsažených v databázi bibliografických citací tedy ve zdrojovém XML dokumentu. Popis základních funkcí by mohl být následující (využití popisovaných celků bude lépe viditelné v ukázce zdrojového kódu v závěru této kapitoly):

- **Atributy elementu xsl:stylesheet:**

Tab. 2.7: Atributy elementu xsl:stylesheet pro výstupní *.tex soubor

Atribut	Hodnota	Popis
xmlns:xsl	"http://www.w3.org/1999/XSL/Transform"	deklarace transformačního souboru
version	"2.0"	určení verze XML technologie

- **Atributy elementu xsl:output:**

Tab. 2.8: Atributy elementu xsl:output pro výstupní *.tex soubor

Atribut	Hodnota	Popis
method	"html"	základní atribut definující html výstup
encoding	"utf-8"	určení typu kodování
omit-xml-declaration	"yes"	zajišťuje správnou strukturovanost při využití kódování typu UTF

- **Využití elementu xsl:template a xsl:apply-template:**

Tato část je rovněž jako u XSL šablon pro *.bib a *.tex soubory základním stavebním kamenem. Konkrétní ukázka je uvedena na konci této kapitoly.

- **Využití elementů a atributů pro formátování výstupního html souboru:**

```
<html>
  <head>
    <title>
      <xsl:text>Databáze bibliografických citací</xsl:text>
    </title>
  </head>
  <body>
    ...
    ...
    ...
  </body>
</html>
```

Dále je použit následující element a jeho atributy:

```
<div align="center">
<div align="left">
<div align="right">
```

Tento element řadí vypisovaný text v závislosti na použitých attributech (na střed, vlevo, vpravo). Dalším využitým elementem je:

```
<strong><font size="+3">Články</font></strong>
```

Element strong říká, že vypsaný text bude uveden tučně a velikost fontu je nastavena pomocí atributu size.

Dále je použito základních párových tagů. K výpisu potřebných textů (znaků) je použit element xsl:text.

- **Využití elementu xsl:sort:**

```
<xsl:sort select="author"/>
```

Tento element seřadí abecedně vzestupně zdroje bibliografických citací v databázi podle elementu author.

- **Využití elementu xsl:if:**

U této šablony se element xsl:if využívá identicky jako v šabloně pro tvorbu výstupních souborů *.tex.

- **Ostatní:** U typu citace electronic (internetové zdroje) bylo pro zobrazení aktivní webové adresy použito následující syntaxe:

```
<xsl:if test="normalize-space(url)">
  URL:
  <a>
    <xsl:attribute name="href">
      <xsl:value-of select="url"/>
    </xsl:attribute>
    <xsl:value-of select="url"/>
  </a>
</xsl:if>
```

Zkrácená ukázka zdrojového kódu pro generování výstupních *.html souborů:

```
<xsl:template match="/">
  <html>
    <head>
      <title>
        <xsl:text>Databáze bibliografických citací</xsl:text>
      </title>
    </head>
    <body>
      <div align="center">
        <strong><font size="+4">Databáze ... citací</font></strong>
      </div>
      <div align="left">
        <xsl:if test="bibliography/book">
          <strong><font size="+3">Knihy</font></strong>
        </xsl:if>
      </div>
      <td>
        <xsl:apply-templates select="bibliography/book">
          <xsl:sort select="author"/>
        </xsl:apply-templates>
      </td>
      ...
      ...
      ...
    </div align="left">
```

```

    <xsl:if test="bibliography/standard">
      <strong><font size="+3">Standardy</font></strong>
    </xsl:if>
  </div>
  <td>
    <xsl:apply-templates select="bibliography/standard">
      <xsl:sort select="author"/>
    </xsl:apply-templates>
  </td>
  <div align="right">
    <font size="-1">This page was created XML Transforma...
  </div>
</body>
</html>
</xsl:template>
...
...
...
<xsl:template match="book">
  <p>
    <xsl:if test="normalize-space(author)">
      <strong>
        <xsl:value-of select="author"/>
      </strong>
      <xsl:text>: </xsl:text>
    </xsl:if>
    <xsl:if test="normalize-space(title)">
      <xsl:value-of select="title"/>
      <xsl:text>. </xsl:text>
    </xsl:if>
    ...
    ...
    ...
    <xsl:if test="normalize-space(isbn)">
      ISBN -
      <xsl:value-of select="isbn"/>
      <xsl:text>.</xsl:text>
    </xsl:if>
  </p>

```

```
</xsl:template>
```

```
...
```

```
...
```

```
...
```

2.3 Ukázky výstupních souborů

Veškeré zobrazené výstupy byly generovány pomocí grafického rozhraní vytvořeného pro parser Saxon, popsaného v kapitole 2.4.

2.3.1 Ukázka výstupu souboru *.bib

V rámci tohoto výstupního souboru nebylo potřeba provádět řazení jednotlivých atributů, protože režim práce s bibliografickými citacemi se definuje pomocí deklarace v prostředí sázecího programu \LaTeX (viz. kapitola 1.3). Jednotlivé atributy typů citací byly ovšem řazeny identicky jako by byl soubor tvořen pomocí programu JabRef. Ukázka výstupního bib. souboru je vidět na obrázku 2.2. Pro formátování databáze bibliografických citací je využit příkaz v prostředí \LaTeX a to:

```
\bibliographystyle{czechiso}
```

Styl czechiso odpovídá normě ČSN 690.

2.3.2 Ukázka výstupu souboru *.tex převedeného na soubor *.pdf

V tomto výstupním souboru jsou atributy jednotlivých citací řazeny a odděleny dle normy ČSN 690. Ukázka výstupu ve formátu pdf je uvedena na obrázku 2.3 a 2.4.

2.3.3 Ukázka výstupu souboru *.html

V tomto výstupním souboru jsou atributy jednotlivých citací řazeny a odděleny dle normy ČSN 690. Ukázka výstupu ve formátu html je uvedena na obrázku 2.5 a 2.6.


```

% This file was created XML Transformation by Ladislav Mazac
% Encoding: Cp1250

@BOOK(xml_upl_zac,
  title = {XML pro úplné začátečníky},
  publisher = {Computer Press},
  year = {2002},
  author = {Grusová Lucie},
  address = {Praha},
  edition = {Vydání první},
  isbn = {80-7226-697-7}
)

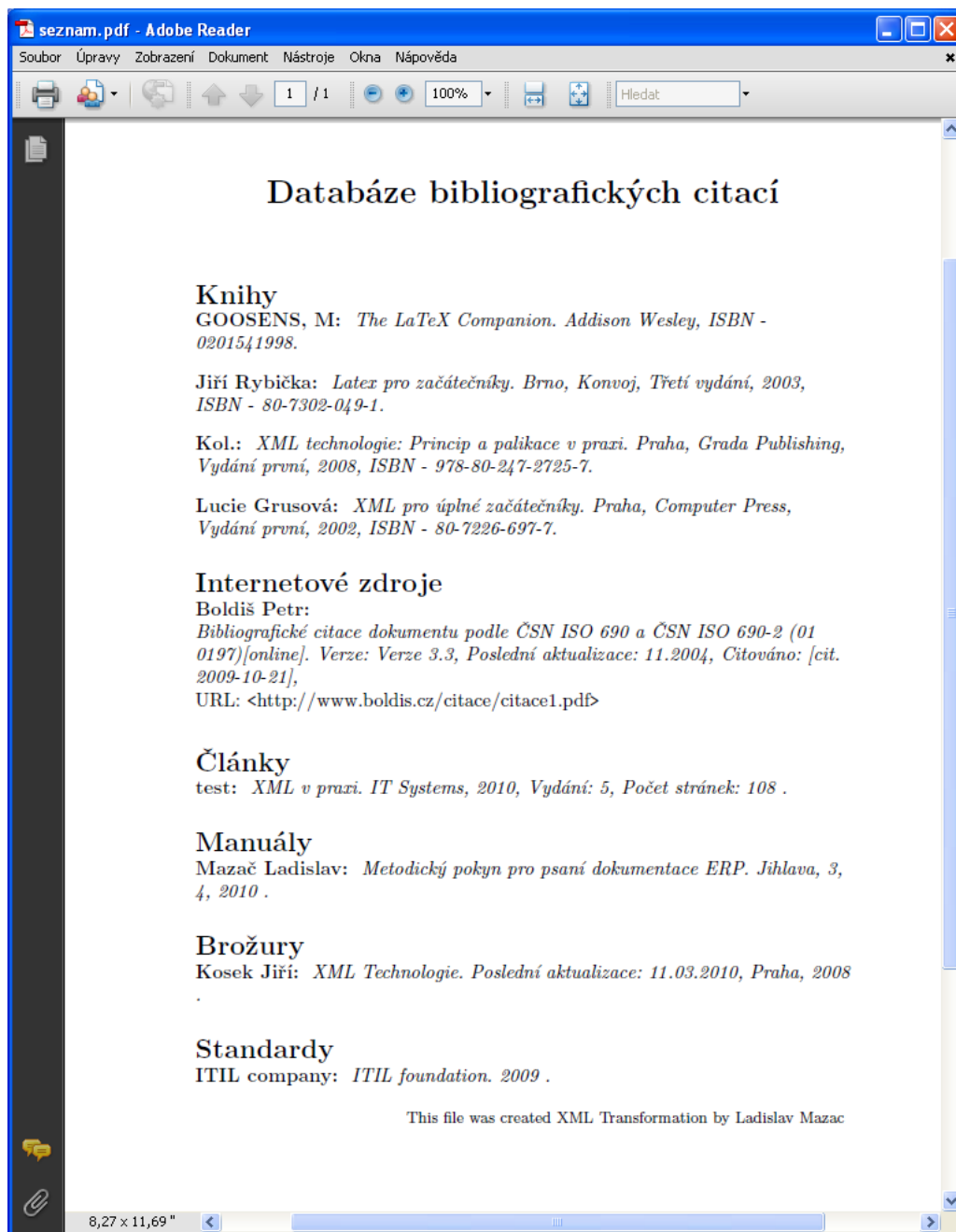
@BOOK(latex_pro_zac,
  title = {Latex pro začátečníky},
  publisher = {Konvoj},
  year = {2003},
  author = {Rybička Jiří},
  address = {Brno},
  edition = {Třetí vydání},
  isbn = {80-7302-049-1}
)

@BOOK(xml_tech,
  title = {XML technologie: Princip a palikace v praxi},
  publisher = {Grada Publishing},
  year = {2008},
  author = {Kol.},
  address = {Praha},
  edition = {Vydání první},
  isbn = {978-80-247-2725-7}
)

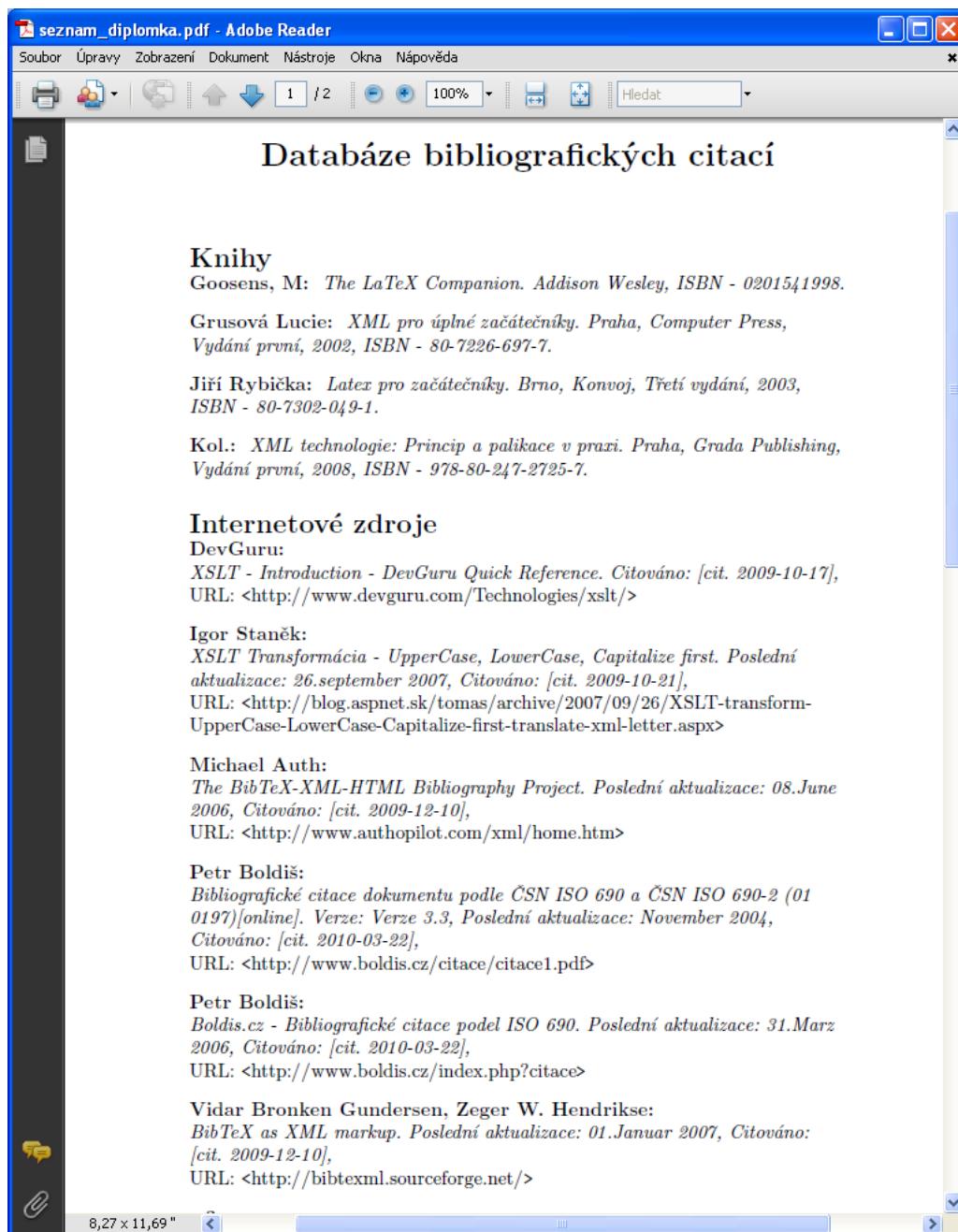
@BOOK(latex_comp,
  title = {The LaTeX Companion},
  publisher = {Addison Wesley},

```

Obr. 2.2: Ukázka výstupního bib souboru použitého v diplomové práci



Obr. 2.3: Ukázka výstupního pdf souboru vygenerovaného ze souboru tex se všemi typy citací



Obr. 2.4: Ukázka výstupního pdf souboru vygenerovaného ze souboru tex pro citované zdroje v této diplomové práci



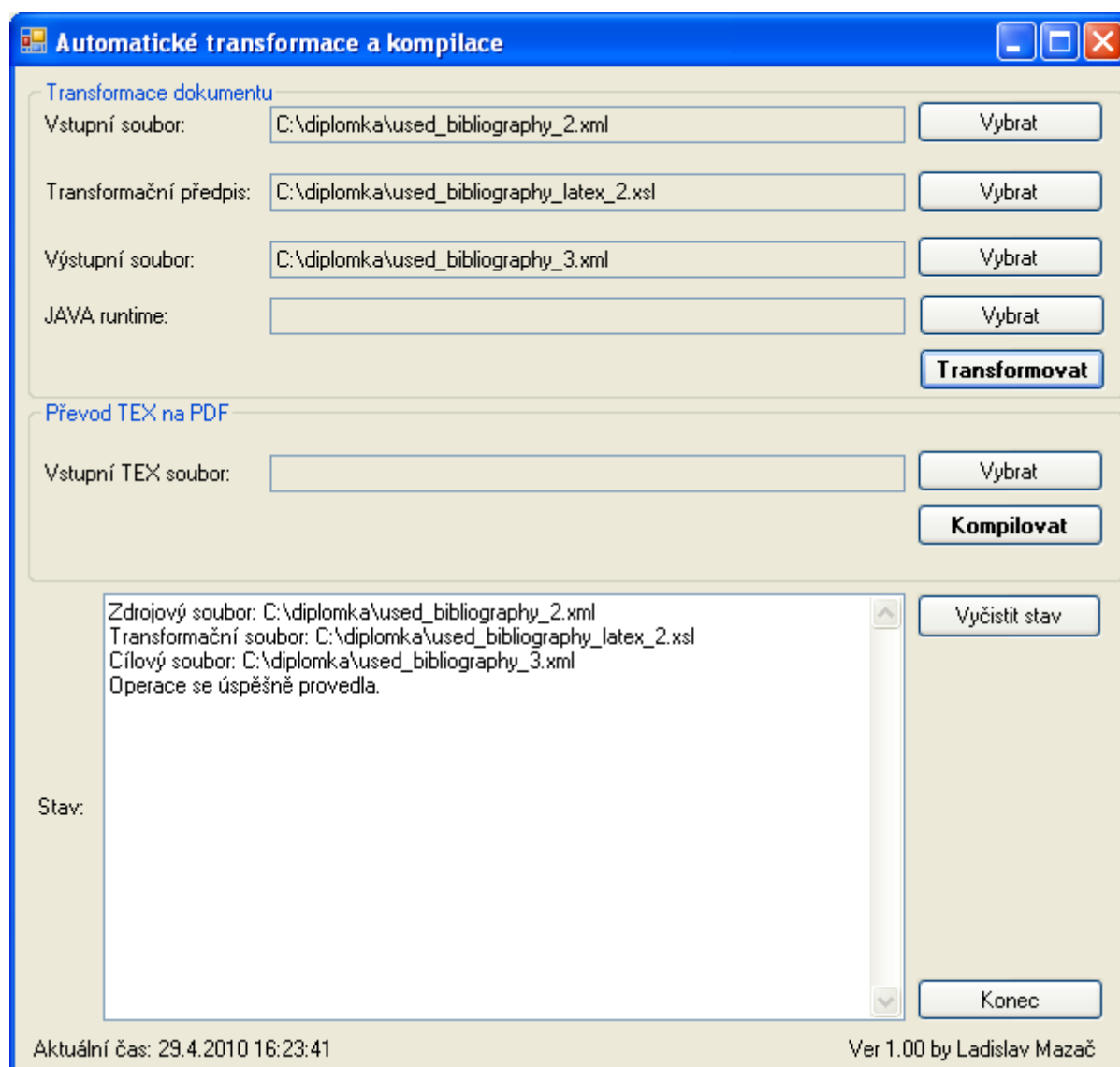
Obr. 2.5: Ukázka výstupního html souboru se všemi typy citací



Obr. 2.6: Ukázka výstupního html souboru pro citované zdroje v této diplomové práci

2.4 Grafický program pro práci s parserem Saxon

Pro usnadnění práce je v rámci této diplomové práce naprogramováno grafické prostředí pro ovládání parseru Saxon9he, jehož hlavní okno je uvedeno na obrázku 2.7. V kapitole 2.1 je uvedena syntaxe, která se zapisuje do příkazového řádku pod adresářem, kde je umístěn samotný parser. Program je naprogramován na technologické platformě .NET v prostředí Visual Studio v jazyce C#. Vytvoření tohoto programu nebylo stěžejní částí diplomové práce, proto zde není popisován zdrojový kód, ale pouze funkčnost a postup práce v programu.



Obr. 2.7: Ukázka výstupního html souboru

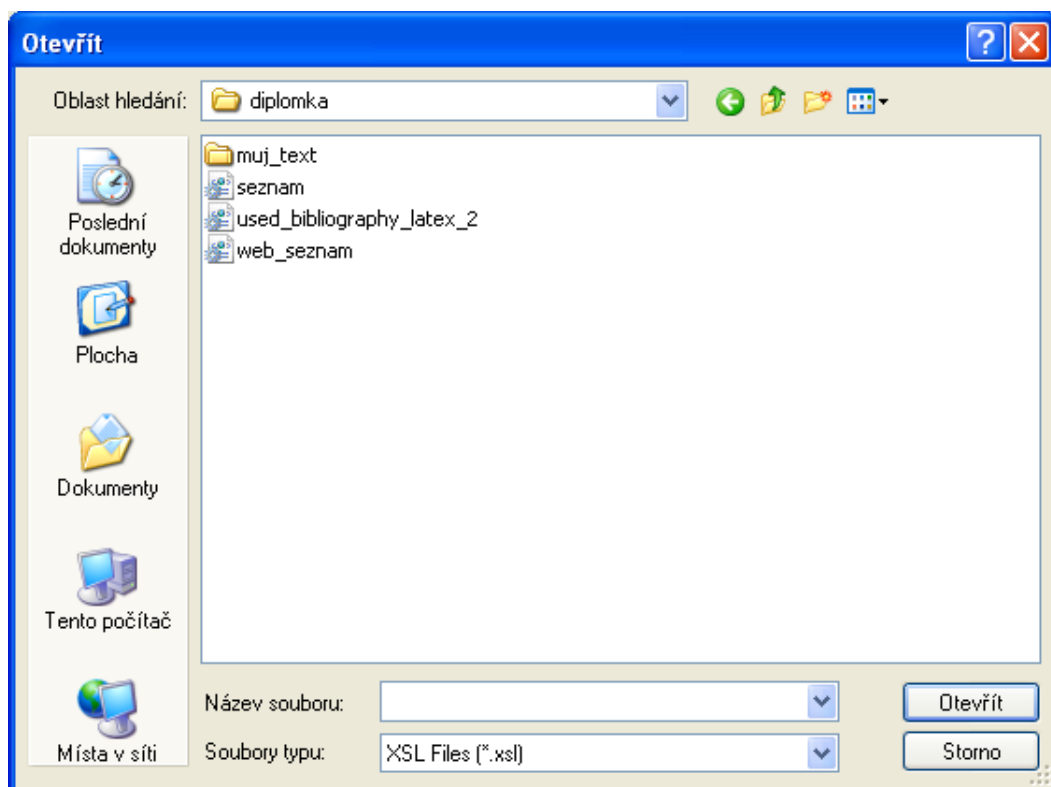
2.4.1 Základní práce v programu a jeho popis

Základní myšlenkou programu je, aby uživatel pro tvorbu výstupního souboru (*.bib, *.tex, *.html) nemusel ukládat soubory do adresáře obsahujícího parser Saxon9he a nemusel vypisovat cestu do příkazové řádky. Samotný parser je v aplikaci natvrdo zapuštěn a v případě problémů s parserem lze pomocí pole s názvem "Runtime Java" nastavit cestu přímo z lokálních diskových jednotek.

Srdcem programu je vyskládání a spuštění příkazu v příkazové řádce. Načtením souboru v grafickém programu se do příkazu pro příkazovou řádku vloží absolutní cesta vybraného souboru. Až je vybrán zdrojový XML soubor, transformační předpis (soubor s příponou xsl) a je určen výstupní soubor, po stisku tlačítka "Transformovat" se provede spuštění vyskládaného příkazu. Výběry souboru probíhají pomocí tlačítka "Vybrat" a u všech výběrů vypadají stejně jako na obrázku 2.8. Zásadní rozdíl je však v tom, že pro takzvané "Vstupní soubory" je výběr omezen pouze na soubory s příponou xsl a pro "Transformační předpis" je výběr omezen pouze na soubory s příponou xsl. V poli "Výstupní soubor" lze vytvořit výstupní soubory s příponami bib, tex, html, xml. Také lze v programu načíst soubory *.tex, které lze pomocí stisku tlačítka "kompilovat" převést na soubor *.pdf. Je využito příkazu pro kompilaci pdflatex název texového souboru. Ovládání programu pro vytvoření výstupního souboru je uvedeno v části s názvem "Transformace dokumentu" a oblast pro vytvoření pdf souboru ze zdrojového tex souboru je v části s názvem "Převod TEX na PDF".

Pro úspěšné provedení transformace je nutné, mít u vstupního XML souboru uložen také jeho DTD soubor. Soubory, které jsou pomocí tohoto programu vybrány, nesmí mít ve své absolutní cestě obsaženu mezeru, jinak program nahlásí chybu a vytvoření výstupního souboru se neprovede. Pokud nejsou vybrány všechny potřebné soubory pro transformaci, při stisku tlačítka program upozorní, který soubor je třeba ještě vybrat.

Program obsahuje okno s názvem "Stav", ve kterém se vypisují ladící hlášky programu, které jsou uvedeny v tabulce 2.9: V případě využívání aplikace se mohou vyskytnout jisté chyby. Aplikace nebyla speciálně laděna a testována, ale byla pouze využívána při tvorbě výstupních souborů uvedených v této diplomové práci. Nesmírná výhoda je, že pokud pracujeme na některém ze souborů a změníme jej, stačí ho pouze uložit a pro vytvoření výstupu stisknout buď tlačítko pro transformaci nebo kompilaci.



Obr. 2.8: Ukázka výběru transformčního předpisu

Tab. 2.9: Ladící a chybové hlášení programu

Úkon	Výpis
Zrušení výběru	"Byl zrušen výběr zdrojového souboru", "Byl zrušen výběr transformačního souboru", "Byl zrušen výběr cílového souboru"
Potvrzení výběru	"Zdrojový soubor a absolutní cesta", "transformační soubor a absolutní cesta", "Cílový soubor a absolutní cesta"
Potvrzení transformace, kompilace	"Operace se úspěšně provedla"
Výpis chybových hlášení	Vypisuje veškeré chybové hlášení a důvody proč transformace nebo kompilace neproběhla v pořádku

3 ZÁVĚR

XML technologie se využívají stále častěji v různých oblastech světa informačních technologií. V dnešní době elektronických dokumentů tvořených pomocí nejrůznějších nástrojů vznikla potřeba implementovat bibliografické citace do programu $\text{BIB}\text{T}\text{E}\text{X}$ ze zdrojového XML dokumentu. V rámci této diplomové práce bylo primárním úkolem vytvořit databázi bibliografických citací. Datovým zdrojem byl dokument XML a jeho předpis byl definován v DTD dokumentu. Pomocí XSL transformací byly vytvořeny výstupní soubory (*.bib, *.html, *.tex, *.pdf). Vytvořený soubor *.bib byl využit v rámci tvorby diplomové práce v sázecím programu $\text{L}\text{A}\text{T}\text{E}\text{X}$. Databáze bibliografických citací vygenerovaná ze zdrojového XML souboru umí uložit a převést na požadované výstupní soubory citace typu kniha (book), článek (article), internetové zdroje (electronic), manuál (manual), brožura (booklet), standard (standard). Nad rámec diplomové práce bylo vytvořeno grafické rozhraní pro snazší ovládání parseru Saxon9he.

Transformační šablony jsou ošetřeny proti vzniku redundatních dat ve výstupních souborech. Pokud nebude vstupní soubor XML obsahovat určité atributy, případně celé typy citací, nebude se do výstupního souboru přenášet jejich formátování. Zadání diplomové práce považuji za zcela splněné a funkční. Důkazem je tato diplomová práce, kde je soubor použitý pro program $\text{BIB}\text{T}\text{E}\text{X}$ vygenerován právě pomocí technologie XML.

LITERATURA

- [1] Auth, M.: The BibTeX-XML-HTML Bibliography Project. [cit. 2009-12-10].
URL <http://www.authopilot.com/xml/home.htm>
- [2] Boldiš, P.: Bibliografické citace dokumentu podle ČSN ISO 690 a ČSN ISO 690-2 (01 0197)[online]. [cit. 2010-03-22].
URL <http://www.boldis.cz/citace/citace1.pdf>
- [3] Boldiš, P.: Boldis.cz - Bibliografické citace podél ISO 690. [cit. 2010-03-22].
URL <http://www.boldis.cz/index.php?citace>
- [4] DevGuru: XSLT - Introduction - DevGuru Quick Reference. [cit. 2009-10-17].
URL <http://www.devguru.com/Technologies/xslt/>
- [5] Goosens, M.: *The LaTeX Companion*. Addison Wesley, ISBN 0201541998.
- [6] Kol.: *XML technologie: Princip a aplikace v praxi*. Praha: Grada Publishing, vydání první vydání, 2008, ISBN 978-80-247-2725-7.
- [7] Lucie, G.: *XML pro úplné začátečníky*. Praha: Computer Press, vydání první vydání, 2002, ISBN 80-7226-697-7.
- [8] Maňák J, k. R. P.: BibTeX.
- [9] Rybička, J.: *Latex pro začátečníky*. Brno: Konvoj, třetí vydání vydání, 2003, ISBN 80-7302-049-1.
- [10] Staněk, I.: XSLT Transformácia - UpperCase, LowerCase, Capitalize first. [cit. 2009-10-21].
URL <http://blog.aspnet.sk/tomas/archive/2007/09/26/XSLT-transform-UpperCase-LowerCase-Capitalize-first-translate-xml-letter.aspx>
- [11] Vidar Bronken Gundersen, Z. W. H.: BibTeX as XML markup. [cit. 2009-12-10].
URL <http://bibtexml.sourceforge.net/>
- [12] w3c: Character entity references in HTML 4. [cit. 2010-03-16].
URL <http://www.w3.org/TR/html4/sgml/entities.html>
- [13] w3c: W3Schools Online Web Tutorials. [cit. 2009-10-13].
URL <http://w3schools.com/>

- [14] w3c: XSL Transformation (XSLT). [cit. 2009-10-16].
URL <http://www.w3.org/TR/xslt>

SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

DTD	Document Type Definition
HTML	HyperText Markup Language
SGML	Standard Generalized Markup Language
XSD	XML Schema Definition
XML	eXtensible Markup Language
xmlns	XML namespace
XSL	eXtensible Stylesheet Language
XSLFO	eXtensible Stylesheet Language Formatting Objects
XSLT	eXtensible Stylesheet Language Transformations

SEZNAM PŘÍLOH

A	Popis přiloženého datového média	61
A.1	Zdrojové soubory potřebné pro transformace	61
A.2	Elektronická podoba diplomové práce	61
A.3	Zdrojové kódy ke grafickému programu k obsluze parseru Saxon9he .	61

A POPIS PŘILOŽENÉHO DATOVÉHO MÉDIA

K diplomové práci je přiloženo datové médium, na kterém jsou uvedeny veškeré zdrojové kódy.

A.1 Zdrojové soubory potřebné pro transformace

Různé varianty XML souborů obsahující zdrojová data, DTD soubor, XSL šablony pro vytvoření výstupů souborů bib, tex, html a také jsou přiloženy konkrétní výstupy bib, tex, html a pdf.

A.2 Elektronická podoba diplomové práce

Zdrojové kódy vytvořené v programu \LaTeX spolu se souborem `diplomka_literat.bib`, který byl v rámci této diplomové práce vygenerován a použit. Dále je přiložena diplomová práce `Diplomova_prace_Mazac_88552.pdf`.

A.3 Zdrojové kódy ke grafickému programu k obsluze parseru Saxon9he

Je přiložen soubor `Automatic Saxon.exe`, kterým se spouští aplikace popsaná v kapitole 2.4 spolu se všemi zdrojovými kódy.