MASARYKOVA UNIVERZITA V BRNĚ
FAKULTA INFORMATIKY

Systém pro správu publikací Laboratoře NLP

BAKALÁŘSKÁ PRÁCE

Petr Šimkovič

Brno, podzim 2005

Prohlášení

Prohlašuji, že tato bakalářská práce je mým původním autorským dílem, které jsem vypracoval samostatně. Všechny zdroje, prameny a literaturu, které jsem při vypracování používal nebo z nich čerpal, v práci řádně cituji s uvedením úplného odkazu na příslušný zdroj.

Vedoucí práce: Mgr. Pavel Šmerk

Shrnutí

Základní funkcí systému pro správu publikací Laboratoře NLP je získávat a aktualizovat publikace členů laboratoře z Informačního systému MU a poskytovat je pomocí webového rozhraní. Publikace mohou být při tom tříděny podle zvolených kritérií. Dále je také umožněno vyhledávání a vkládání publikací do košíku, ze kterého si uživatel může vygenerovat různé formy výpisu publikací.

Klíčová slova

Perl, XML, XML Schema, XHTML, CGI, PostgreSQL, BibTeX, LaTeX

Obsah

Kapitola 1

Úvod

V dnešní době je používání Internetu součástí života každého běžného člověka. Uživatel jeho pomocí může nalézt mnoho zajímavých informací. Jejich prezentace bývá realizována různými webovými technologiemi, které se neustále vyvíjejí, zdokonalují a nahrazují novějšími verzemi. Při jejich správném vybrání a implementování mohou stránky nabízet mnoho užitečných funkcí, služeb a vlastností.

Cílem práce je implementace systému, který by zpracovával publikace a prezentoval je pomocí vhodného webového rozhraní. To by také mělo umožňovat základní správu systému.

Kapitola 2

Současný stav

Publikace jsou nyní zveřejňovány na serveru laboratoře v české [1] a anglické verzi [2]. K jejich vytváření je použit makrosystém, který pro expanzi maker používá procesor m4. Současné dokumenty jsou statické, a proto tedy neumožňují vyhledávání ani jiné podobné funkce. Samotné publikace jsou uloženy na serveru v adresáři `/nlp/publications` a jsou dostupné na adrese <`http://nlp.fi.muni.cz/publications`>.

Pokud bychom si zobrazili všechny publikace současných i minulých členů laboratoře a prošli bychom si všechny odkazy, přišli bychom k následujícímu zjištění. A sice, že všechny publikace uložené na serveru nejsou odkazovány z tohoto seznamu publikací. Proto také systém obsahuje kontrolu, která ověřuje, zda jsou jednotlivé publikace v IS MU odkazovány právě do tohoto adresáře.

---

1. <`http://nlp.fi.muni.cz/nlp/aisa/NlpCz/Publikace_clenu_laboratore.html`>
2. <`http://nlp.fi.muni.cz/nlp/NlpEn/Publications_of_laboratory_members.html`>

Kapitola 3

Databáze

Dnešní webové aplikace využívají služeb různých databází. Mezi jejich nesporné výhody patří zejména transakční zpracování a následná konzistence dat, jednoduchý přístup k datům a množství dalších funkcí, které sama databáze nabízí. Všechny tyto výhody bychom těžko implementovali, kdybychom data uchovávali jen v nějakém souborovém systému. Nutno podotknout, že databázi zabírá k zajištění všech těchto služeb určitý čas, i když je nepatrný. Zejména jde o paralelní zpracování a autorizaci přístupu.

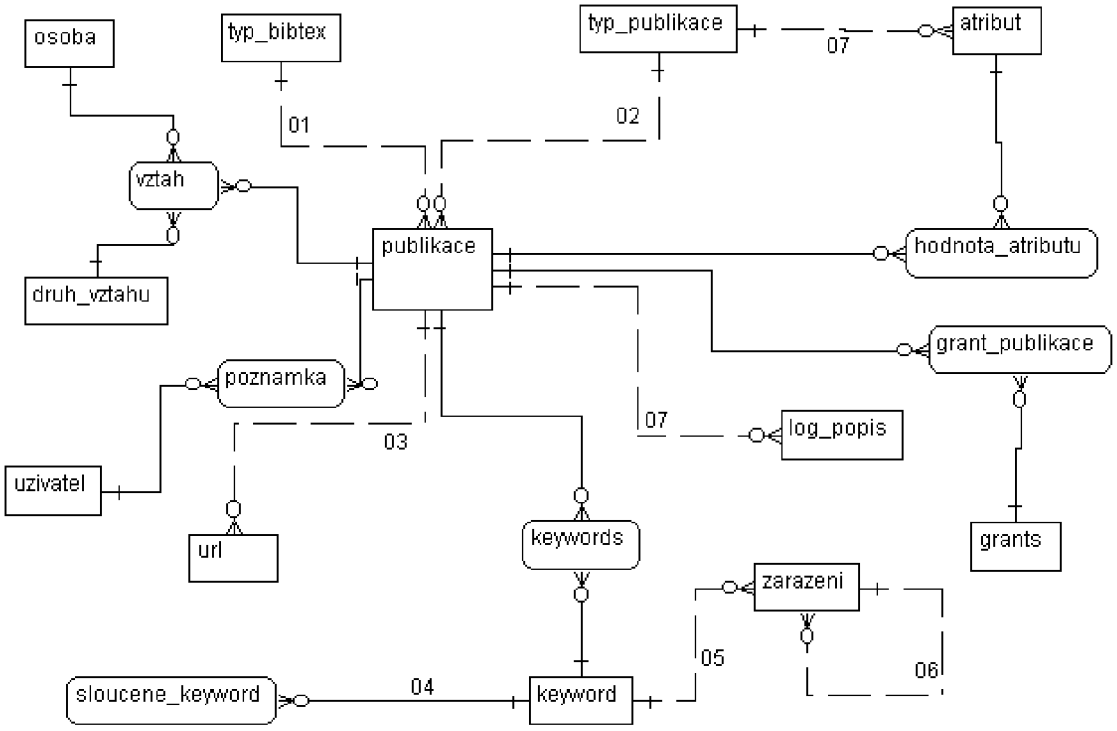
Výběr tvořily tři databáze: MySQL, Oracle a PostgreSQL, které jsou přístupné na serveru *db.fi.muni.cz*. Databázový server společnosti Oracle patří mezi nejlepší světové databázové servery. Jde však o komerční záležitost, a tak jsem vybíral mezi zbylými dvěma databázemi. Databázový open source server MySQL je oblíbený zejména při používaní webových aplikací. Je velice rychlý, má dobrou podporu standardů jazyka SQL, ale v dřívějších verzích postrádal některé vlastnosti, například transakce, vnořené SELECTy, uložené procedury a pohledy. Dnes už jsou tyto nedostatky převážně napraveny, ale bude jistě nějakou dobu trvat, aby vše bylo zoptimalizováno a bez chyb. Také server PostgreSQL je open source. Vyniká stabilitou, rychlostí, dobrou podporou a dobrou integrací pokročilých technologií. Primárně je určen pro použití v UN*Xových OS. Mimo jiné nabízí vlastní tvorbu triggerů, vestavěných funkcí a transakční zpracování. Vše patří mezi požadavky správného a efektivního běhu našeho systému, a tak jsem vybral právě tuto databázi.

3.1 Logický datový model

3.1.1 Entity

Název: atribut
Typ: kernel
Objektem typu (#atribut) je každý druh informace libovolné publikace, který není společný pro všechny publikace. Poznámka: např. období vzniku bakalářské práce.

Název: druh_vztahu
Typ: kernel
Objektem typu (#druh_vztahu) je každý druh vztahu mezi osobou a publikací, který má smysl definovat. Poznámka: např. autor, editor nebo vedoucí práce.



Obrázek 3.1: Logický datový model

Název: grants
Typ: kernel
Objektem typu (#grants) je každý udělený grant, který má smysl evidovat.

Název: grant_publikace
Typ: kernel
Objektem typu (#grant_publikace) je každá reprezentace vazby mezi grantem a publikací se smyslem:
Existence vazby (#grant_publikace) mezi daným grantem (#grants) a danou publikací (#publikace) / $0,1 : 1,1$. Poznámka: mezi jedním grantem a danou publikací může být jen jeden vztah.

Název: hodnota_atributu
Typ: associative
Objektem typu (#hodnota_atributu) je každá reprezentace vazby mezi atributem a publikací se smyslem:
Hodnota (hodnota_atributu) daného atributu (#atribut) dané publikace (#publikace) / $0,1$:

0, M.

Název: keyword
Typ: kernel
Objektem typu (#keyword) je každé klíčové slovo, které může charakterizovat nebo charakterizuje nějakou publikaci.

Název: keywords
Typ: associative
Objektem typu (#keywords) je každá reprezentace vazby mezi klíčovým slovem a publikací se smyslem:
Existence vazby (#keywords) mezi daným klíčovým slovem (#keyword) a danou publikací (#publikace) / $0, 1 : 1, 1$. Poznámka: mezi jedním klíčovým slovem a danou publikací může být jen jeden vztah.

Název: log_popis
Typ: kernel
Objektem typu (#log_popis) je každý popis publikace, který byl v minulosti změněn. Poznámka: umožňuje nám logovat změny popisu publikace.

Název: osoba
Typ: kernel
Objektem typu (#osoba) je každá osoba, která je nebo může být autorem publikace či vedoucím práce nebo se pro ni stahují publikace z ISu MU.

Název: poznamka
Typ: associative
Objektem typu (#poznamka) je každá reprezentace vazby mezi publikací a uživatelem se smyslem:
Poznámka (#poznamka) daného uživatele (#uzivatel) k dané publikaci (#publikace) / $0, M : 0, M$.

Název: publikace
Typ: kernel
Objektem typu (#publikace) je každá kniha, článek, práce nebo jiná publikace, kterou je účelné v systému evidovat.

Název: sloucene_keyword
Typ: kernel
Objektem typu (#sloucene_keyword) je každé klíčové slovo, které bylo v minulosti sloučeno v jiné.

Název: typ_bibtex
Typ: kernel
Objektem typu (#typ_bibtex) je každý definovaný bibtexový typ publikace, který používá

program BibTEX. Poznámka: např. MISC, BOOK aj.

Název: typ_publikace
Typ: kernel
Objektem typu (#typ_publikace) je každý typ publikace, který je používán vlastním systémem. Poznámka: např. publikace z NLP, cizí publikace aj.

Název: url
Typ: kernel
Objektem typu (#url) je každý odkaz publikace na jiný dokument nebo jiné URL.

Název: uzivatel
Typ: kernel
Objektem typu (#uzivatel) je každý uživatel systému mající svůj login. Poznámka: největší význam přináší, pokud jsou stránky chráněné souborem .htaccess.

Název: vztah
Typ: associative
Objektem typu (#vztah) je každá reprezentace vazby mezi osobou publikací a druhem vztahu se smyslem:
Existence daného druhu vztahu (#druh_vztahu) mezi danou osobou (#osoba) a danou publikací (#publikace) / $0, M : 0, M$.

Název: zarazeni
Typ: kernel
Objektem typu (#zarazeni) je každá pozice klíčového slova v hierarchii klíčových slov.

3.1.2 Vazby

Ref: 01
Bibtexový typ (#typ_bibtex), kterého je daná publikace (#publikace) / $1, 1 : 0, M$.

Ref: 02
Typ (#typ_publikace), kterého je daná publikace (#publikace) / $1, 1 : 0, M$.

Ref: 03
Odkazy (#url)-s dané (#publikace) / $0, M : 1, 1$.

Ref: 04
Nová hodnota (#keyword) v minulosti sloučeného klíčového slova (#sloucene_keyword) / $1, 1 : 0, M$.

Ref: 05
Pozice (#zarazeni) klíčového slova (#keyword) v hierarchii klíčových slov / $0, M : 1, 1$.

Ref: 06
Rodič (#zarazeni) jiného zařazení (#zarazeni) / $1, 1 : 0, M$. Poznámka: Tato vazba umožňuje

řadit klíčové slova do struktury podobné stromu, kde má každý uzel svého rodiče. Tzv. kořen má hodnotu klíčového slova NULL.

Ref: 08
Změna popisu (#log_popis) publikace (#publikace) / 0, M : 1, 1. Poznámka: vazba umožňuje logovat změny popisu všech publikací.

3.2 Přístup k databázi

Celý systém je implementován v jazyku Perl, a tak je používán modul *DBI*, Database Interface. Modul poskytuje metody pro připojení k databázovému systému, metody pro příkaz, dotaz a získání výsledku dotazu. Podporuje také transakční zpracování. Samotné připojení je realizováno postupem, který je vidět na následujícím příkladu.

```
my $dbh = DBI->connect("dbi:Pg:dbname=$dbname;host=db.fi.muni.cz;
                        port=$port","$login","$password",
                        {RaiseError=>1, AutoCommit=>0})
                                or die 'chyba pri spojeni s databazi';
```

Tento příklad zajistí spojení s databázovým systémem, kde proměnná `$dbname` představuje jméno schématu, `$port` číslo portu, `$login` uživatele, `$password` heslo. První volba `RaiseError` s hodnotou 1 nám zajistí vypsání chyby databáze pomocí die (RaiseErorr), např. při zpracování dotazu. Konečně volba `AutoCommit` s hodnotou 0 zajišťuje, že každý příkaz nebude následován implicitním commitem. Pokud připojení proběhlo bez chyb, v proměnné `$dbh` bude databázový handler pro poslání příkazů.

Kapitola 4

Charakteristika systému

Kvůli častému zpracování textu je implementace systému provedena ve skriptovacím jazyce Perl. Tento jazyk byl také vybrán pro velké množství modulů, které sám obsahuje nebo jsou dostupné z CPANu (Comprehensive Perl Archive Network, <www.cpan.org>). Nejpoužívanější v tomto systému je modul CGI, dále jsou používány moduly pro změnu kódování, XML validaci aj. Systém také obsahuje vlastní modul, kde jsou především deklarovány konstanty pro cesty k souborům nebo adresářům.
Při návrhu systému byl brán zřetel na to, aby bylo možné do systému přidávat nové funkcionality, případně je odebírat nebo měnit. Hlavní činnost spočívá v prezentování jednotlivých publikací, které jsou rozděleny na publikace členů laboratoře, diplomové či bakalářské práce a na ostatní publikace.

Kapitola 5

Části systému

Implementace systému musela splňovat několik požadavků, které byly na systém kladeny. Jednalo se především o aktualizaci publikací a jejich způsob prezentování. Systém byl navržen jako celek skládající se z osmi částí, které využívají společných zdrojů – databáze.

5.1 Aktualizace publikací

První část představuje aktualizaci publikací členů laboratoře a kontrolu nad všemi publikacemi v systému. Do systému také vkládá nové publikace. Při každé aktualizaci se kontroluje, zda nebyla publikace v ISu nějak změněna. Pokud ano všechny údaje se opraví až na anotace a klíčová slova. Ty zůstávají stejné a jejich změna je možná pouze pomocí systému.

5.2 Publikace členů Laboratoře NLP

Druhá část se stará o českou a anglickou prezentaci publikací, jejichž autory jsou pouze samotní členové laboratoře. Vedle dynamického výpisu existují dvě verze statického výpisu. První z nich vzniká přímo z IS MU, druhá se generuje z předem aktualizované databáze. Hlavní rozdíl spočívá v tom, že autoři mohou být v databázi upravováni, a tak tento výpis může tedy být korektnější.

5.3 Diplomové a bakalářské práce

Třetí část prezentuje a spravuje diplomové či bakalářské práce členů. Ty se vkládají pomocí webového formuláře a posléze se ukládají do databáze. Opět existuje dynamická a statická verze jejich výpisu.

5.4 Cizí publikace

Čtvrtá část umožňuje vkládat cizí publikace, které se nějak týkají oblasti zpracování přirozeného jazyka, a prezentuje všechny předchozí publikace jako celek. Umožňuje také měnit anotace, klíčová slova a popisy jednotlivých publikací a vkládat poznámky uživatelů k příslušným publikacím.

5.5 Modul košík

Zmíněné poslední tři části v sobě zahrnují nástroj, představující pátou část. Jeho pomocí si mohou uživatelé generovat vybrané publikace do třech různých formátů: XHTML stránky, BibTEXového souboru a souboru, který slouží k začlenění do LATEXového dokumentu jako zdroj literatury pomocí prostředí *thebibliography*.

5.5.1 XHTML

Nový formát webových stránek XHTML [?] vychází z XML (eXtensible Markup Language) a je tak vlastně jakousi jeho podmnožinou. Z HTML si přebírá některé názvy značek a z XML pravidla jejich použití. U XHTML se již nehovoří o značkách (tagech), ale o elementech. Ty se rozlišují na párové a nepárové. Párové mají svou počáteční a koncovou značku, např. ... . Nepárové mají jen počáteční, která je sama koncová, např.
. Každý XHTML dokument (kromě dokumentů v kódování UTF-8 nebo UTF-16, kde to není nutné) musí obsahovat deklaraci XML společně s kódováním dokumentu

```
<?xml version="1.0" encoding="iso-8859-2"?>
```

a deklaraci typu dokumentu. Na výběr jsou tři: strict, transitional a frameset.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
 "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

- `Strict` - nejpřísnější typ nedovolující použití jakéhokoliv formátování vzhledu dokumentu. Takový vzhled se nejčastěji utváří pomocí kaskádových stylů CSS.
- `Transitional` - tento typ umožňuje vzhled definovat již v samotném dokumentu
- `Frameset` - typ je totožný s předchozím, ale navíc umožňuje použití rámců. Ty však obecně není doporučováno používat.

Mezi hlavní zásady při psaní XHTML dokumentů patří: Značky se píší malými písmeny, žádné elementy se nesmí navzájem prolínat a všechny počáteční elementy musí mít svůj příslušný ukončovací. Zdrojový kód výpisu košíku do HTML pak tedy vypadá následujícím způsobem.

```
<ul>
 <li><a href="http://is.muni.cz/lide?jazyk=cz;uco=42803">Prýmek Miroslav</a> -
 <a href="http://is.muni.cz/lide?jazyk=cz;uco=1648">Horák Aleš</a>.
 <strong>Design and Implementation of Multi-Agent System for Analysis of
 Electrical Power Networks.</strong> In Proceedings of ELNET 2005. Ostrava,
 ČR : VŠB TU, Ostrava. s.525-538. 1ET400300414, projekt VaV. 2006.
 <a href="http://is.muni.cz/publikace/publikace_simple.pl?jazyk=cz;id=593524">
 Info</a>  </li>
</ul>
```

5.5.2 BIB

BibTEXový formát je určen pro především pro sázecí TEXové systémy. Jako XML má svá pravidla zápisu, má i bibtexový formát svá vlastní. Jednotlivé publikace začínají symbolem „@“, za kterým následuje typ vstupní publikace (nerozlišují se malá a velká písmena). Systém používá hlavně tyto BibTEXové typy:

- `masterthesis` - školní práce (povinné atributy: author, title, school a year)
- `misc` - jakákoliv publikace (žádné povinné atributy)
- `proceedings` - jednání z konference (povinné atributy: title a year)

Každá publikace dále obsahuje klíč pro citovanou publikaci, kterou uvádíme v dokumentu, a jednotlivá pole, která publikaci charakterizují. Je zvykem psát název pole na nový řádek, za něj pak symbol „=“ a do složených závorek napsat hodnotu pole. Jednotlivá pole jsou oddělena čárkou. Jako příklad polí mohou být: address, author, title, year aj. Mnohé verze BibTEXu nestandardní pole ignorují takže k publikaci můžeme zaznamenávat mnohé údaje. Pokud chceme dokument přeložit LATEXem tak, že se při prvním přeložení vytvoří soubor s příponou .aux, který obsahuje instrukce pro program BibTEX. Po té se spustí BibTEX a vytvoří se soubor s příponou .bll obsahující seznam citací, který je nadále zpracovatelný LATEXem. Pro správnost dokumentu je dobré pak ještě dokument dvakrát přeložit, aby se správně vytvořily všechny odkazy. Výsledný bibTEXový soubor může vypadat následovně:

```
@PROCEEDINGS{1,
  author = "Prýmek Miroslav and Horák Aleš",
  title = {Design and Implementation of Multi-Agent System for
           Analysis of Electrical Power Networks},
  year = {2006},
  key = {intelligent agent, power system simulation},
  note = {In Proceedings of ELNET 2005. Ostrava, ČR : VŠB TU,
          Ostrava. s.525-538},
  annote = {In this paper, we describe the first stages ...},
  grant = {1ET400300414, projekt VaV},
  entitle = {Design and Implementation of Multi-Agent System for
             Analysis of Electrical Power Networks},
  cztitle = {Návrh a implementace multiagentního systému pro analýzu
             elektrorozvodných sítí},
  url = {"http://is.muni.cz/publikace/publikace_simple.pl?jazyk=cz;id=593524"}
}
```

5.5.3 BIBITEM

Poslední možností zobrazení publikace z košíku jsou položky bibitem, které slouží jako zdroj literatury rovněž pro TEXové systémy. Narozdíl od předešlého formátu jsou uvedeny ve zdrojovém dokumentu. Samotná literatura je uvedena v prostředí „thebibliography“. Každá publikace začíná makrem \bibitem{}, kde ve složených závorkách je uveden klíč pro citovanou literaturu. Dále se uvádějí údaje o dané publikaci.

```
begin{thebibliography}{99}
\bibitem{1}
{\sc Prýmek Miroslav, Horák Aleš.}
{\em Design and Implementation of Multi-Agent System for Analysis
of Electrical Power Networks.}
In Proceedings of ELNET 2005. Ostrava, ČR : VŠB TU, Ostrava. s.525-538.
Grant: 1ET400300414, projekt VaV.
2006.
\standardhyphens
\href{http://is.muni.cz/publikace/publikace_simple.pl?jazyk=cz;id=593524}
{Info}{ }
\splithyphens
\end{thebibliography}
```

5.6 Kontrola

Tuto část tvoří kontrola různých údajů a je prezentována jako HTML stránka. Nejdříve se provádí kontrola, zda jsou odkazy do adresáře, ve kterém jsou uloženy publikace laboratoře, korektní. Po té se zkontrolují všechny publikace uložené v tomto adresáři, jestli na ně vůbec nějaká publikace z ISu odkazuje. Další dvě kontroly rovněž souvisí s tímto adresářem. První z nich kontroluje, zda odkaz ukazuje přímo na soubor nebo do adresáře, přičemž systém preferuje druhou variantu, tedy odkazy na adresáře. Druhá kontrola se zabývá existencí adresáře TEX pro každou publikace laboratoře. Předposlední kontrola ověřuje správnost odkazů všech publikací v systému. Konečně poslední kontrola se týká osob uložených v databázi. Jednak zda UČO dané osoby patří skutečně této osobě a také jestli k osobě bez vlastního UČO neexistuje osoba se stejným jménem v ISu MU.

5.7 Hierarchie klíčových slov

Šestou část představuje hierarchie klíčových slov. Zde můžeme řadit publikace do hierarchie podle zadaných klíčových slov. V každé části jsou zobrazeny jen publikace, které obsahují jen příslušná klíčová slova.

5.8 Správa klíčových slov a osob

Poslední část umožňuje spravovat osoby a klíčová slova. Tato část byla nutná, protože mnoho publikací obsahuje různé chyby nebo jejich údaje jsou nějakým způsobem duplicitní. Např. jedna publikace obsahuje klíčové slovo animation a druhá animations, nebo existuje jedno klíčové slovo ve dvou různých jazycích. Co se týče osob, existují publikace s autorem jako Jan Zizka a Jan Žižka nebo Petr Sojka (UČO 2378) a P Sojka (UČO 2378) apod. Z tohoto důvodu systém umožňuje slučovat různé osoby nebo různá klíčová slova.

Kapitola 6

Uživatelé systému

Ne všechny části systému jsou určeny každému uživateli. Z tohoto pohledu můžeme systém rozdělit na dvě základní části:

1 Pro první skupinu uživatelů systém nabízí pouze statický a dynamický dokument pouze s publikacemi členů Laboratoře NLP. Je zde i možnost statického dokumentu s bakalářskými či diplomovými pracemi.

2 Druhá skupina uživatelů může využívat všech částí systému. Tedy i dynamického dokumentu diplomových či bakalářských prací a také seznamu, ve kterém jsou umístěny všechny publikace uložené systému. Do tohoto seznamu lze vkládat nové publikace a měnit požadované údaje.

Je nutné zmínit, že takto jsou uživatelé rozděleni až při existenci souboru `.htaccess` v příslušném adresáři, jak je uvedeno v dokumentaci. Při rozdělení se také mění způsob práce s poznámkami. Především to znamená, že měnit danou poznámku může jen její vlastník (uživatel, který poznámku vložil). Mění se také výpis poznámek, kdy jsou pro každého uživatele vypisovány jen jeho poznámky. Při způsobu práce s klíčovými slovy, anotací a popisem nenastává žádná změna. Tyto údaje jsou společné pro všechny uživatele.

Kapitola 7

Použité technologie

7.1 JavaScript

Pro zviditelnění některých údajů o publikaci bylo použito JavaScriptu. Je to klientský skriptovací jazyk, který je interpretován prohlížečem, aniž by bylo nutné kontaktovat server. Může být začleněn do samotné stránky nebo do samostatného souboru, jako je u našeho systému. V současné době je podporován všemi hlavními prohlížeči. Je nutné si uvědomit, že Java a JavaScript jsou dva různé programovací jazyky. Podobná je pouze syntaxe. Poskytuje řadu objektů, ale sám není zcela objektový. Může pouze používat již vytvořené objekty, kterých je omezené množství. Sem patří objekty: Window, Document, String, Date a Math. Tohoto jazyka spolu s kaskádovými styly je využito právě pro snadné zobrazování anotací a klíčových slov. Pro zobrazení těchto položek, které jsou implicitně nezobrazované, nemusí uživatel kontaktovat server. Prohlížečům, které JavaScript nepodporují, slouží volba pro zobrazení klíčových slov a anotací pomocí formulářového pole CGI skriptu – checkboxu. Aby skript fungoval i ve starších prohlížečích, používají se funkce, zajišťující kompatibilitu práce s objekty v různých browserech.

```
function ukaz(id){
    getObj(id).display = (getObj(id).display == "block")? 'none' : 'block';
}
```

Tato funkce zobrazí nebo skryje element, který je určen parametrem `id`. Samotný parametr je v našem případě atribut id elementu div příslušného CGI skriptu.

```
function zobraz_k(){
    var form = document.getElementById("form");
    var checkbox = form.getElementsByTagName("input");
    var zobrazit = (checkbox[5].checked)? "block" : "none";
    var elCont = document.getElementById("vypis");
    var colLists = elCont.getElementsByTagName("div");
    for (var i = 0; i < colLists.length; i++) {
        var elList = colLists[i];
        if (elList.className == "key") {
           elList.style.display = zobrazit;
        }
    }
}
```

Tato funkce je poněkud složitější a slouží k zobrazení klíčových slov všech publikací. Nejprve si do proměnné `form` přiřadíme formulář s atributem id s hodnotou *form*. Toho využijeme pro získání pole checkboxů tohoto formuláře do proměnné `checkbox`. Víme, že checkbox pro zobrazení je šestý v pořadí, a tak můžeme do proměnné `zobrazit` přiřadit hodnotu *block*, pokud je checkbox zaškrtnut nebo hodnotu *none* v opačném případě. Podobně jako checkboxy získáme všechny elementy div v elementu s hodnotou atributu id *vypis*, což je tabulka publikací, a následně všechny tyto divy procházíme. Pokud mají atribut class nastaveny na hodnotu *key* (elementy div s klíčovými slovy), nastavíme pomocí kaskádových stylů vlastnost display na příslušnou hodnotu. Obě tyto funkce jsou vyvolány pokud klikneme na příslušný checkbox formuláře pro výpis publikací.

7.2 CSS

Většina skriptů používá pro zobrazení jednotlivých prvků kaskádové styly. Výhody tohoto způsobu spočívají v oddělení obsahu textu od jeho formátování. Můžeme tak používat jeden styl pro více dokumentů a také změnit vzhled bez zásahu do samotného skriptu. CSS (Cascading Style Sheets) neboli kaskádové styly představují sadu pravidel pro uspořádání a zobrazení dokumentu. Nejčastěji se používají pro HTML, XHTML a XML dokumenty. O jejich standardizaci se stará konzorcium W3C [1].

```
div.anotace, div.key, div.popis, div.poznamky {
   margin-left: 10px;
   margin-bottom: 10px;
   margin-top: 10px;
   padding: 5px;
   background-color: #e5e5e5;
   width: 95%;
   white-space: normal;
   border: 1px solid #a3a3a3;
}
```

Tato ukázka představuje zobrazení anotací, klíčových slov, popisů a poznámek. Definuje vnější odsazení (margin) těchto elementů, podobně vnitřní odsazení (padding), pozadí, šířku, způsob zobrazení bílých znaků a rámečku kolem těchto elementů. Pokud bychom si výpis publikací chtěli vytisknout, definujeme způsob zobrazení těchto elementů následovně.

```
div.key, div.anotace, div.poznamky, div.popis {
    margin-left: 5%;
}
```

Zde určujeme jen vnější odsazení zleva, aby byl výpis přehlednější. Výsledek můžeme sami vidět, pokud si stránku zobrazíme jako náhled tisku v našem prohlížeči.

---

1. <http://www.w3.org/Style/CSS>

7.3 CGI

Aby bylo umožněno dynamické vyhledávání a třídění, bylo použito pro zobrazení publikací CGI skriptů. Obecně tyto skripty fungují tak, že se provedou na straně serveru a výsledek se pošle na klientský prohlížeč. CGI skripty mohou být napsány v různých programovacích jazycích jako je Perl, může se použít předem zkompilovaných programů, např. napsaných v jazyce C, nebo je můžeme napsat v nějakém shellu. Častou výhodou použití zkompilovaných programů je rychlejší odezva na straně serveru. Každý CGI skript, který je napsán ve skriptovacím jazyce, musí na svém prvním řádku určovat, jak se bude interpretovat. Dále musí následovat hlavička pro WWW server a prázdný řádek pro oddělení obsahu.
Vstup dat do skriptu je umožněn dvěma metodami - POST a GET. U první metody se parametry předávají pomocí proměnné *QUERY_STRING*. Jednotlivé parametry jsou odděleny apostrofem nebo někdy středníkem.

```
http://nlp.fi.muni.cz/.../uprav_pole.cgi?co=popis;bib=1;text=pdf
```

Výhodou je, že uživatel zasílá data v uváděném URL a také že odezva serveru je mnohem rychlejší. Uvádění dat v URL může být na druhé straně i nevýhodou, např. pro zajištění bezpečnosti. Pro větší objemy dat je tato metoda nevhodná kvůli omezení délky zadaného URL ze strany webových prohlížečů.
Druhá metoda získává data ze standardního vstupu, čímž odstraňuje omezení velikosti dat. Skript využívající této metody může použít proměnné *CONTENT_LENGTH*, která obsahuje celkovou velikost dat v bytech.
Skripty systému jsou napsány v jazyce Perl a využívají obou metod pro předávání dat do skriptu a také modul CGI [2] , který je vhodný pro tvorbu různých formulářů. Tento modul nabízí téměř všechny HTML elementy včetně elementů potřebných pro tvorbu formuláře. Pomocí něho je například snadné stáhnout uživatelem zadaný soubor, zpracovat jej a výsledek poslat do prohlížeče uživatele. Základní potřebou je vytvoření nového CGI objektu a hlavičky dokumentu.

```
#!/usr/bin/perl -w
use CGI qw(:standard);
use CGI::Carp qw/fatalsToBrowser/;
use strict;
my $q = new CGI();
print $q->header(-charset=>'iso-8859-2'),
        start_html(-lang=>'cz',-title=>'Title');
```

Podobně jako se na ukázce vytváří hlavička a titulek stránky, vytváří se tak i ostatní HTML elementy, např. element nebo <form>. Při psaní skriptů bylo použito *use CGI* pro jejich ladění na příkazové řádce a *use CGI::Carp qw(fatals_to_browser)* pro zobrazení chybových hlášek do prohlížeče.

2. <http://search.cpan.org/~lds/CGI.pm-3.10/CGI.pm>

7.4 XML

Následující dvě technologie slouží v systému zejména pro vkládání nových publikací, které se nějak týkají zpracování přirozeného jazyka. XML je metajazyk pro vytváření značkovacích jazyků. Přesněji se jedná o standard nebo spíše doporučení konsorcia W3C. Základním požadavkem kladeným na každý XML dokument je, že musí být dobře utvořen (well-formed). Toto nastane, právě když:

1. Obsahuje prolog (hlavičku) a právě jeden, tzv. kořenový element. Dále může obsahovat komentáře. Ty se zapisují následovně:

   ```
   <!-- text komentáře -->
   ```

   Prolog se píše na první řádek dokumentu, např.:

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

   Pokud se prolog (hlavička) neuvede předpokládá se kódování UTF-8 nebo UTF-16.

2. Musí vyhovovat všem pravidlům pro správné utvoření uvedeným ve specifikaci.

3. Totéž platí pro každou analyzovanou (parsovanou) entitu přímo nebo nepřímo odkazovanou v dokumentu. Především to znamená, že každý počáteční element musí být uzavřen svým ukončovacím elementem a dále, že jednotlivé elementy se nesmějí prolínat.

7.5 XML Schema

Hlavní význam schémat spočívá pro definici značkovacích jazyků. Vlastní XML Schema nám umožňuje přesně popsat, jak má daný XML dokument vypadat a můžeme ho tedy i validovat. To je ověřit, zda dokument splňuje všechna omezení popsaná v tomto schématu. XML Schema umožňuje definovat obsah jednotlivých atributů nebo elementů pomocí datových typů a tak aplikace nemusí nutně pracovat jen s řetězci, jak může být u XML zvykem. Podrobnější popis tohoto schématu a jak má správně XML dokument pro vkládaní publikací vypadat, je již popsán v samotném systému a tak jej tady, i pro svou složitost, neuvádím.

Kapitola 8

Závěr

V rámci práce se mi podařilo implementovat systém pro správu publikací Laboratoře NLP. Jeho součástí je webová aplikace, která nabízí prezentaci publikací, jejich vkládání, úpravu a různé formy jejich výpisu. Navíc umožňuje jednotlivé publikace vyhledávat a třídit je podle zvolených parametrů. V současné době si můžeme funkčnost systému vyzkoušet na této adrese <http://nlp.fi.muni.cz/~xsimkov/nova/hlavni.htm>.
Při práci jsem narazil na několik menších problémů, které se týkají zpracování textu. Díky tomu, že je systém napsán ve skriptovacím jazyce Perl, který poskytuje dostatečné prostředky pro zpracování textu, se mi podařilo všechny problémy vyřešit. Rovněž jsem uvedl všechny použité webové technologie, které byly při práci použity.

Literatura

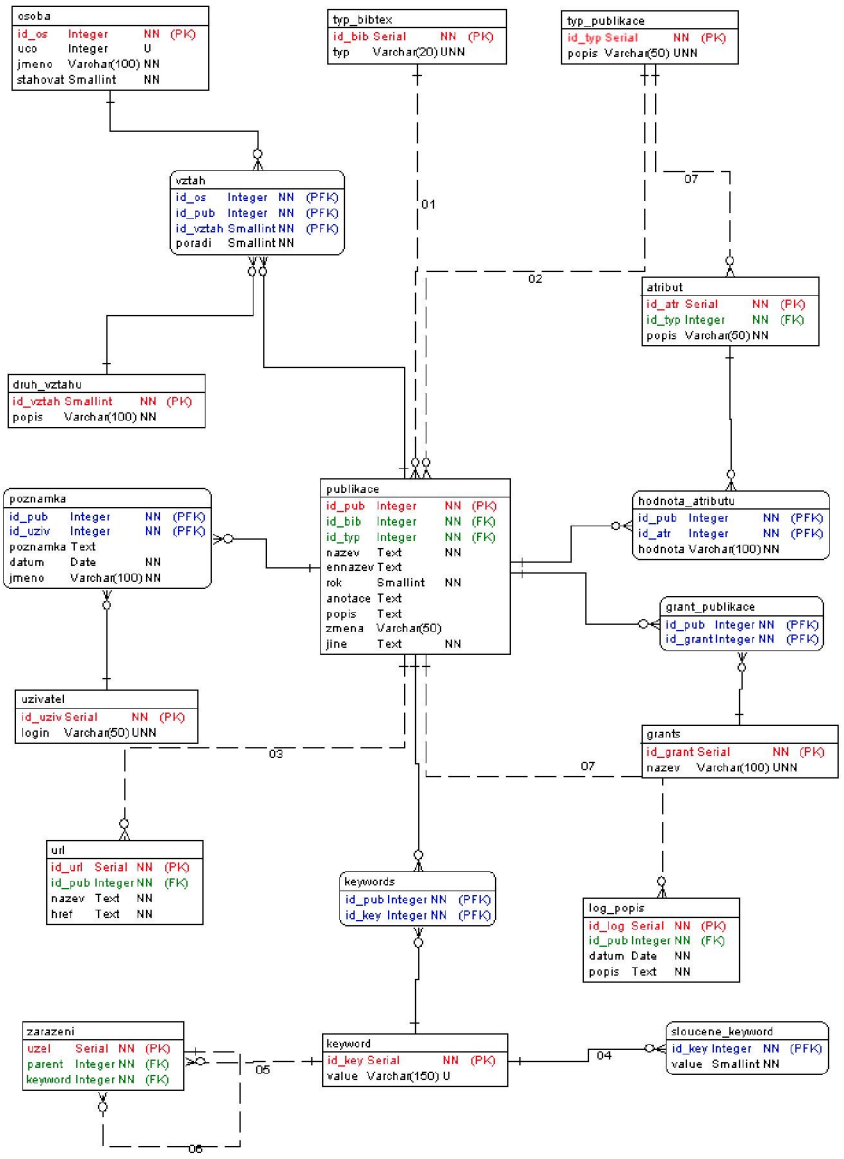
[1] Mikle, P.: , *XCSS - referenční příručka*, Zoner Press, 2004, 80-86815-13-7, Copyright © 2004 ZONER software s.r.o..

[2] Rybička, J.: , *LaTeX pro začátečníky*, KONVOJ, 2003, 80-7302-049-1, Copyright © 2003 Jiří Rybička.

[3] Mikle, P.: , *XDHTML - referenční příručka*, Zoner Press, 2004, 80-86815-01-3, Copyright © 2004 ZONER software s.r.o.. 5.5.1

Dodatek A

Fyzický datový model



Dodatek B

Dokumentace

B.1 Požadavky na provoz systému

Celý systém byl vyvinut pod operačním systémem Linux a je také pro něj primárně určen. Dále je nutné mít nainstalovaný interpret jazyka Perl s modulem DBI::Pg a také mít zprovozněného démona *cron*. Dále systém využívá databázi PostgreSQL a je dobré ji mít zkompilovanou tak, aby správně uměla české třídění a používala kódování ISO-8859-2.

B.2 Instalace

Dekomprimací souboru `publikace.tar.bz2` získáme dva adresáře: `publikace_scripts` a `publikace_html`. V prvním musíme nejdříve změnit soubor `MyConstant.pm` – cestu ke skriptům, HTML souborům a proměnné pro spojení s databází. Podobně také soubory `run.sh`, `static_prace.sh` a `static_publikace`.
Pro vytvoření databáze nám slouží skripty, které se nacházejí v podadresáři `sql`. Nejprve se musíme spojit s databází, např. pomocí konzole následujícím způsobem:

```
$ psql -h databazovy_server jmeno_databaze uzivatel
```

Po zadání hesla se můžeme dostat to tohoto podadresáře například takto:

```
\cd cesta_k_adresari_sql;
```

Potom jen stačí vytvořit potřebné tabulky a funkce:

```
\i create_db.sql;
\i create_fce.sql;
```

Druhý adresář se musí nacházet, tak aby byl přístupný pro webové prohlížeče a jiné aplikace. Zde stačí jediná změna souboru `MyCGI.pm` v jeho podadresáři `cgi-bin`. Vše je dobře zakomentováno, takže není nutné nějakého podrobnějšího popisu.
Pokud zde ještě vytvoříme v podadresáři `hidden` pro cgi skripty skrytý soubor `.htaccess`, můžeme mít přístup k těmto skriptům pouze pomocí autorizace konkréktního uživatele, jak umožňuje server Apache.

B.3 Démon cron

Cron je systémová utilita, která umí spouštět jiné programy v předem definovanou dobu nebo v požadovaném intervalu. K zajištění aktualizace a stahování publikací z ISu MU stačí do něj přidat soubor *run.sh*, který se nachází v adresáři *publikace_scripts*.
Přidat aplikaci do crona můžeme dvěma způsoby. První způsob spočívá k nakopírování souboru do některého z adresářů */etc/cron.daily*, */etc/cron.hourly*, */etc/cron.weekly* nebo */etc/cron.monthly*. Z názvů by mělo vyplývat, jak často by se měla aplikace spouštět. Druhý způsob je o něco složitější, ale dovoluje mnohem více komfortu. Stačí na příkazové řádce zavolat.

```
crontab -e
```

Otevře se seznam úloh, které má již uživatel v cronu umístěny. K přidání stačí vyplnit šest parametrů oddělených mezerou nebo tabulátorem, které znamenají:

1 - minuta

2 - hodina

3 - den v měsíci

4 - měsíc

5 - den v týdnu (0 - neděle, 1 - pondělí, ...)

6 - cesta k aplikaci

Místo parametru lze napsat hvězdičku znamenající, že parametr není definován.

```
0 1 * * 1 cd "adresar se souborem run.sh";./run.sh
```

Tento příklad spouští skript *run.sh* každé pondělí v 1:00. Samozřejmě je nutné uvést celou cestu k tomuto skriptu. Jestliže bychom nuly na místě minuty zapsaly hvězdičku, skript by se spouštěl každou minutu první hodiny každého pondělí.