

ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE

Fakulta elektrotechnická – Katedra počítačů

**DIPLOMOVÁ PRÁCE**

Rozšíření databáze publikací

Prohlašuji, že jsem svou diplomovou práci vypracoval samostatně a použil jsem pouze podklady (literaturu, projekty, SW atd.) uvedené v příloženém seznamu.

Udělují souhlas s užitím tohoto školního díla ve smyslu § 60 zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon).

V Teplicích dne 21.1.2009

Podpis

## Anotace

Tato práce je pokračováním implementace databáze publikací, kterou jsem navrhl v r. 2006 za účelem uchovávání publikací ve formátu pdf/ps a následného vyhledávání a formátování referencí na publikace. Zaměřuji se zde na zdokonalení uživatelského rozhraní a implementaci nové funkčnosti jako je např. fulltextové vyhledávání, export/import z/do systému BibTeX, správa uživatelů atd.. Součástí této práce je jak diskuze nad možností použití podobných, již existujících systémů, tak i jejich srovnání s aplikací, kterou jsem sám implementoval.

## Abstract

This master thesis is an extension of the original publications database, which I designed in 2006 in order to save publication documents and consequently to search and format the publication references. I focus on improving the user interface and the implementation of new features such as fulltext search, export/import to BibTeX system, managing user rights, etc. This thesis also contains a discussion about the option of using already existing systems and their comparsion with the aplication implemented by myself.

# Obsah

1	Úvod a souhrn požadavků .....	1
2	Analýza.....	3
2.1	Uživatelské role .....	3
2.2	Datové požadavky .....	6
2.3	Procesní požadavky .....	8
2.4	Importy a exporty .....	11
2.5	Použité technologie .....	12
3	Zhodnocení existujících systémů .....	13
3.1	Open Source systémy .....	13
3.1.1	Aigaion .....	13
3.1.2	Webový systém BibAdmin.....	14
3.2	Komerční systémy .....	15
3.2.1	Biblioscape .....	15
3.2.2	EndNote .....	15
3.2.3	WriteNote .....	16
3.2.4	Webový portál CiteSeer.....	16
3.2.5	ACM Portal .....	16
3.2.6	RefWorks.....	17
4	Implementace.....	19
4.1	Datový model .....	19
4.2	Autentizace a autorizace .....	25
4.3	Implementace katalogu publikací.....	27
4.3.1	Fulltextové vyhledávání .....	28
4.3.2	Vyhledávání podle jmen autorů.....	29
4.3.3	Zobrazení detailu publikace: .....	29
4.3.4	Generování citací podle normy ČSN a IEEE .....	31
4.4	Implementace rozhraní pro zadavatele.....	34
4.5	Implementace importu ze systému BibTeX .....	38
4.6	Implementace administračního rozhraní .....	42
4.6.1	Formulář pro modifikaci publikace .....	42
4.6.2	Nové záložky .....	44
5	Zabezpečení .....	45
5.1	Autentizace uživatelů .....	45
5.2	Ošetření uživatelských vstupů.....	45

5.3	Chráněné adresáře.....	46
6	Testy .....	47
7	Porovnání s existujícími systémy .....	47
8	Závěr.....	48
9	Seznam literatury .....	49
10	Přílohy .....	51
10.1	Uživatelská a instalační příručka.....	51
10.1.1	Instalace .....	51
10.1.2	Přihlašování .....	52
10.1.3	Rozhraní čtenářů.....	53
10.1.4	Rozhraní pro zadavatele .....	57
10.1.5	Administrační rozhraní.....	60
10.2	Obsah přiloženého CD .....	63
10.3	Rozdělení publikací a jejích atributů v systému BibTeX.....	0

# 1 Úvod a souhrn požadavků

Cílem této práce je rozšířit stávající webovou databázi publikací [1] tak, aby v ní bylo možno vyhledávat podle klíčových slov a zajistit fulltextové vyhledávání v přiložených PDF nebo PS souborech. Dále bylo požadavkem rozšířit databázi publikací o exporty do standardizovaných formátů IEEE a ČSN ISO 690. Funkcí této databáze by také měl být export a import z/do formátu BibTeX. Původně se uvažovalo i o importu/exportu do XML a importech přes protokol Z39.50, což se však ukázalo v průběhu analýzy jako nevhodné. Stěžejním ze všech požadavků mělo být celkové zlepšení vzhledu a uživatelské přívětivosti webové aplikace.

Původní databáze publikací, kterou jsem implementoval v r. 2006 jako svou bakalářskou práci měla za cíl fungovat jako jednoduchý systém, do kterého bude možno zadávat vědecké publikace a vyhledávat je podle názvu a tématu, ke kterému se vztahují. Aby šlo publikace zobrazovat ve standardních citačních formátech a bylo možno k nim přikládat přílohy v podobě dokumentů PDF.

Požadavky na zlepšení uživatelského rozhraní, které vznikly během používání aplikace, byly následující:

- Řazení záznamů. Záznamy v databázi, které se zobrazují na webu, je potřeba řadit podle několika kritérií. Doposud fungovalo řazení pouze u některých záložek administračního rozhraní. V první řadě bylo zapotřebí umožnit řazení publikací při jejich vyhledávání a zobrazení výsledků vyhledávání alespoň podle názvu a roku vydání.
- Vyhledávání publikací dle autorů. Vyhledávání dle autorů, tak jak fungovalo doposud, bylo poněkud nedorozumělé a tedy použitelné pouze v omezené míře. Vyhledávání by mělo fungovat tak, že uživatel do vstupního pole napíše jméno nebo seznam jmen autorů, namísto staré varianty, kdy uživatel vybíral jednoho autora pomocí rozbalovacího seznamu.
- Vyhledávání podle autorů udělat více interaktivní pomocí našeptávače s použitím technologie AJAX. Využití našeptávače by bylo u nového způsobu vyhledávání velmi vhodné. Uživateli, který vyplňuje pole autorů, aplikace sama nabízí autory, kteří již v databázi figuruje. A on má možnost z nabídky vybírat.
- Zavést pořadí autorů. Během provozu aplikace se ukázalo, že toto je důležitá funkcionality, která staré implementaci silně chybí. Pořadím autorů při citování referencí obvykle chceme zdůraznit jejich důležitost, co se týče jejich vztahu k dané publikaci. Obvykle dáváme na první místo autora, který se největší měrou přičinil na jejím vzniku.
- Vylepšit formulář pro zadávání publikace tak, aby se zadávání autorů dalo provádět lépe než doposud a aby bylo možno přiřadit jim pořadí. Toto se týká i předchozího požadavku. Výběr autorů – jak doposud fungoval, neumožňoval určení pořadí a nebyl vhodný pro výběr několika autorů najednou. Byl použit prvek list-box s vlastností multiselect. Výběr několika autorů najednou se provádělo myší za současného stisku klávesy Ctrl. V dlouhém seznamu tak nebylo možno přehledně najít vybrané položky a uživatel často nechtěně výběr autorů zrušil tím, že uvolnil klávesu Ctrl.
- Zrušit povinnost některých vlastností publikace, zejména vadila povinnost zadat autory. U některých publikací autory neznáme.
- Zavést více typů publikací. Stará aplikace rozlišovala pouze 4 typy publikací,

takže v některých případech bylo třeba publikaci přiřadit typ „ostatní“ a nebylo možno zachytit všechny její vlastnosti do standardních polí. Jediné řešení bylo použít tzv. rozšířené atributy [1, strana ...]

- Editace entit, které se k publikaci vážou (např. vydavatel...), už při zadávání publikace.
- Hromadné schvalování publikací v administračním rozhraní. Vylepšení by mělo administrátorovi usnadnit práci tím, že schválí všechny publikace najednou, aniž by je musel po jedné otevírat.
- Možnost zobrazení iniciálů autora v citaci zkratkou. Některé formáty citací předepisují, že jména autorů mají být zobrazena zkratkou, jen příjmení celé.
- U citací s více autory než je povolená mez mít možnost zobrazovat pouze prvního autora. Toto se týká zejména formátu ČSN ISO 690.
- U publikací evidovat více vlastností než uměla stará aplikace. Vhodné je zavedení např. polí abstract, url atd.
- Umožnit uživatelům vkládat osobní a globální poznámky k článkům.
- Převést aplikaci do angličtiny.

Další funkcionalitou, kterou bylo potřeba zavést je správa uživatelů aplikace a autorizace podle oprávnění, která jim přísluší. Stará aplikace už obsahovala jakýsi základ, který ovšem nebyl implementován tak, aby se dala autorizace používat.

Autentizaci bylo potřeba vylepšit tak, aby se zabezpečil přenos hesel po síti, stará aplikace používala obyčejnou autentizaci http basic.

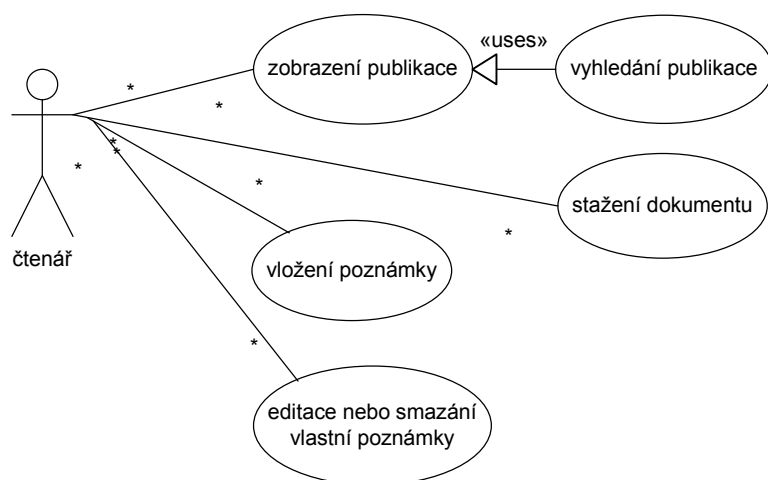
## 2 Analýza

### 2.1 Uživatelské role

Uživatelské role bylo vhodné oproti starému způsobu poněkud upravit. Stará aplikace rozlišuje 3 typy rolí (administrátor, zadavatel a uživatel), viz [1, kapitola 15.1.1], přičemž role zadavatele a uživatele se prolínají. Aplikace nebyla schopná zjistit, kterou roli autentizovaný uživatel zaujímá.

Připomeňme, že uživatel měl mít pravomoce k prohledávání databáze a stahování pdf dokumentů. Zadavatel mohl zadávat publikace a administrátor měl „všechna“ zbývající oprávnění. V původním pojetí se obě role prolínají - zadavatel hrál také roli uživatele a naopak. Vhodnější se ukázalo tyto 2 role od sebe oddělit a podle nich autorizovat. Původní roli „uživatel“ budu nyní nazývat vhodnějším výrazem „čtenář“.

Následující diagram ukazuje případy užití pro roli „čtenář“:



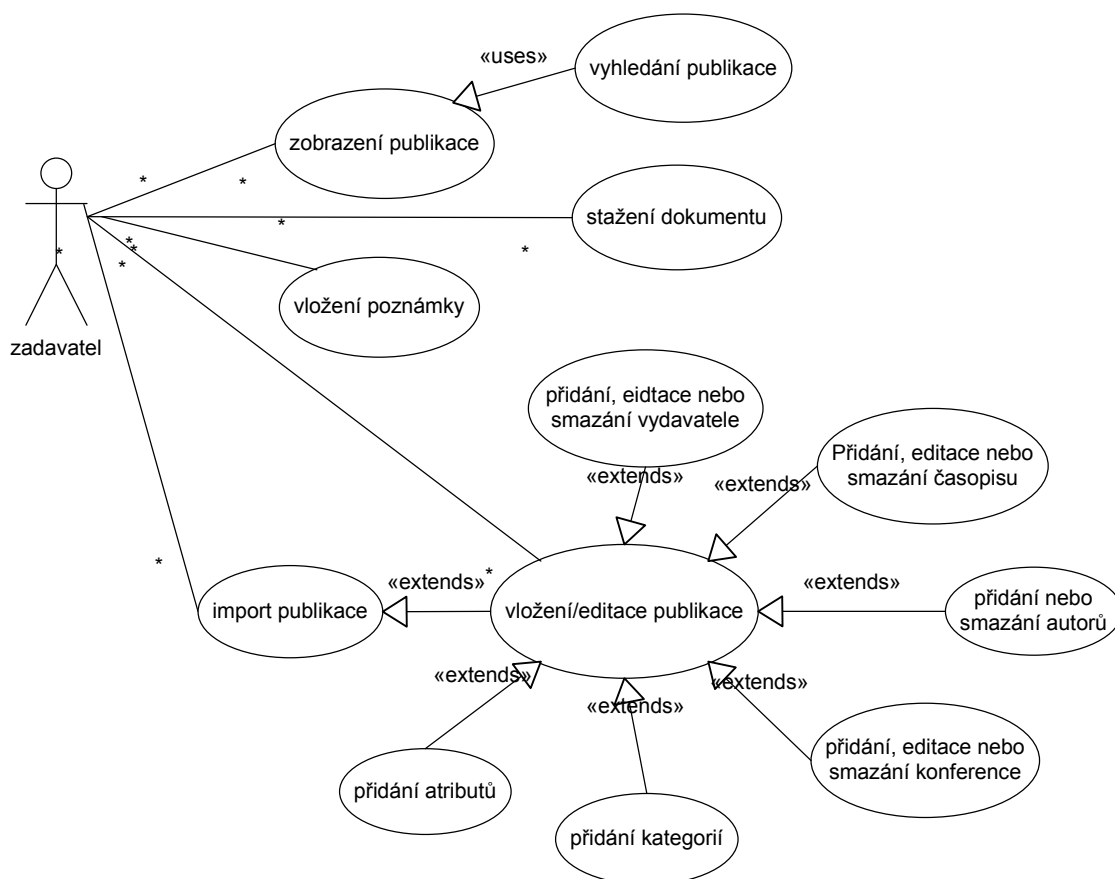
Obrázek 1 - Případy užití pro roli čtenář

Uživatel zadá kritéria pro vyhledávání. Podle nich se zobrazí seznam nalezených publikací, může některou publikaci vybrat a zobrazit její popis, stáhnout dokument, který k publikaci patří, stáhnout BibTeX definici publikace a zobrazit citaci ve zvoleném formátu.

Poznámku, kterou uživatel vkládá, by měla být dvojího typu - osobní nebo globální. Osobní poznámku by měl vidět pouze uživatel, který ji zadal. Globální poznámka by měla být viditelná pro všechny.



Role „zadavatel“ v sobě zahrnuje i roli „čtenář“:

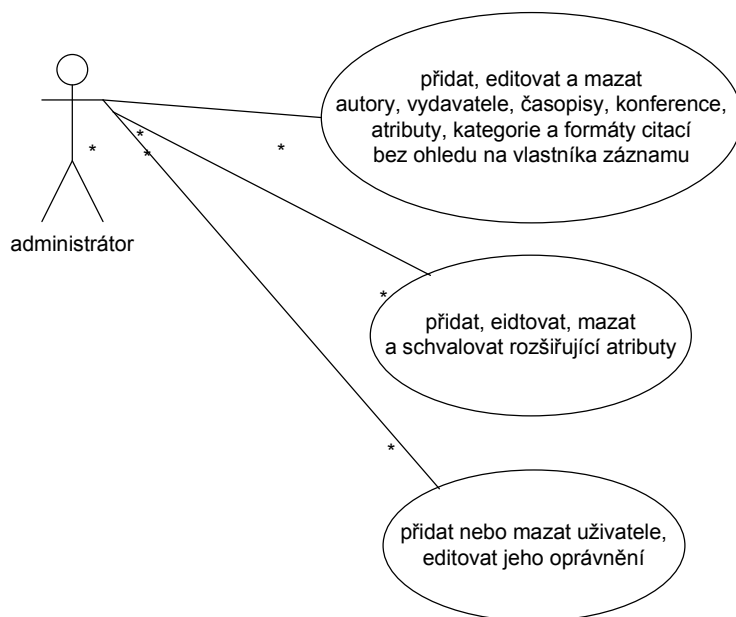


Obrázek 2 - Případy užití pro roli zadavatel

Role „zadavatel“ má mít možnost kromě přidávání publikací, také publikace editovat. Má mít možnost mazat a editovat entity, které se na publikaci vážou. Sem patří konference, časopisy, vydavatelé a autoři. Je však třeba určitým způsobem zajistit, aby jejich změnami neovlivnil negativně další publikace, které jsou na těchto entitách závislé. V žádném případě by nemělo být možné, aby zadavatel mohl měnit jména autorů (vydavatelů atd.), kteří jsou již přiřazeni k nějakým publikacím. Tím by mohl v databázi způsobit nenávratné škody. Totéž platí pro jejich mazání. Mazání by mělo být povoleno pouze u těch záznamů, které nebyly zatím asociovány s žádnými publikacemi, a ještě lépe by bylo kontrolovat kdo záznam do databáze vložil a posléze dovolit mazání záznamu pouze tomu člověku, který je jeho vlastníkem.

U těchto entit se však budou evidovat ještě další doplňkové údaje, které jsou nepovinné, a jejich pozdější úpravou se nenaruší významová správnost dat v publikacích. Povolení úprav těchto záznamů všem zadavatelům má smysl. Této problematice se budu věnovat v kapitole 1.2.

Role „administrátor“ v sobě zahrnuje pravomoce zadavatele i čtenáře, pro jednoduchost uvádím zjednodušený diagram, který znázorňuje pouze případy užití, které má role administrátor navíc:



**Obrázek 3 - Případy užití pro roli administrátor**

Administrátor by měl mít právo editovat všechny záznamy, které se na publikace vážou bez ohledu na to, který člověk je založil nebo zda jsou již přiřazeny existujícím publikacím. Administrátor má mít možnost upravovat a zakládat nové uživatele aplikace a přidělovat jim příslušná oprávnění. Dále by měl mít možnost editovat a schvalovat rozšiřující atributy, stejně jak tomu bylo u původní aplikace [1, kapitola 2.2.1].

Oproti původní aplikaci v této nové verzi upouštím od procesu schvalování viz [1, kapitola 2.2.3]. Proces schvalování publikací se ukázal být v praxi zbytečně restriktivní a do budoucna by vadil.

Navrhuji, aby publikace nově zadaná do databáze byla ihned viditelná a bylo možno ji zobrazit. Totéž by mělo platit i pro záznamy, které se k publikaci vážou (kategorie, autoři, vydavatelé, konference a časopisy). Tyto záznamy by měly být hned viditelné všem ostatním zadavatelům. Navrhuji ovšem, aby rozšiřující atributy musely i nadále procházet procesem schvalování.

## 2.2 Datové požadavky

### *Typy publikací:*

Používáním původní databáze publikací se přišlo na to, že už nevystačíme se 4 druhy publikací, i když zde byla možnost používat tzv. rozšiřující atributy, které jsem navrhl v rámci své bakalářské práce právě pro tento účel. Bylo nutno navrhnout jemnější dělení a podle něj evidovat i další vlastnosti, které může publikace mít. Zejména co se týče základního požadavku, aby aplikace dokázala import a export do formátu BibTeX jsem se rozhodl typy publikací a jejich vlastnosti převzít z něj.

Systém BibTeX dělí publikace na 13 typů a pro každý typ jsou definována povinná a volitelná pole [2]. Přehlednou tabulku těchto polí a jejich přiřazení typům publikací uvádím v příloze v kapitole 10.3

### *Typy publikací, jak je dělí systém BibTeX:*

#### @article

Novinový článek, nebo článek z časopisu.

#### @book

Kniha s jednoznačně identifikovatelným vydavatelem.

#### @booklet

Brožura. Publikace, která je vytištěna a svázána, ale je bez identifikace vydavatele.

#### @inbook

Část knihy, kterou může být buď kapitola, nebo text vymezený rozsahem stran.

#### @incollection

Část knihy mající svůj vlastní název.

#### @inproceedings

Článek z konference.

#### @manual

Technická dokumentace.

#### @misc

Ostatní.

#### @mastersthesis

Diplomová práce.

#### @phdthesis

Disertační práce.

#### @proceedings

Sborník prací.

#### @techreport

Report publikovaný školou nebo jinou institucí.

@unpublished

Dokument, který má autora i název, ale není publikován.

#### *Konference a časopisy:*

Dále bylo zapotřebí založit číselník „konference“. Při zadávání publikací totiž často zadáváme části sborníků, které vznikly na nějaké konferenci, a není moc pohodlné ani vhodné zadávat např. názvy konferencí a místo konání znovu a znovu. Místo toho by bylo lépe, kdyby při zadávání publikace šlo již vybrat konkrétní konferenci ze seznamu. Publikace může vyniknout pouze na 1 konferenci, proto vztah mezi konferencí a publikací je 1:N. U konferencí nás zpravidla zajímá:

- název
- rok
- datum konání od – do
- místo konání
- ISBN sborníku

přičemž název konference spolu s rokem určují konferenci unikátně. Kvůli tomu, že unikátnost konference určují 2 pole.

Stejně tak je vhodné založit číselník „časopis“. K jedné publikaci má smysl přiřazovat max. 1 časopis, relace mezi časopisem a publikací je 1:N. U časopisů nás zajímá:

- název
- ISSN
- volume
- číslo

#### *Poznámky:*

Pro jednu publikaci můžeme evidovat více poznámek. Vztah mezi entitami publikace a poznámkou je 1:N. U poznámek k publikacím má smysl evidovat:

- text poznámky
- příznak osobní/globální
- vlastníka poznámky

#### *Dokumenty:*

Jak jsem již naznačil v úvodu, důležitou součástí vyhledávání v databázi publikací je fulltextové vyhledávání v samotném textu publikací. Textový obsah pdf/ps dokumentů je zapotřebí archivovat v databázi spolu s identifikací publikace, ke které se tento text vztahuje. Otázkou bylo, zda bude publikace mít k sobě přiloženo více nebo jen jeden soubor pdf/ps. Nakonec jsme společně s vedoucím práce usoudili, že bude vhodné mít možnost přiložit k publikaci jeden i více souborů. Při fulltextovém prohledávání se bude brát v úvahu text, který vznikl spojením všech přiložených dokumentů a výsledkem hledání bude (budou) konkrétní publikace.

### *Uživatelé aplikace:*

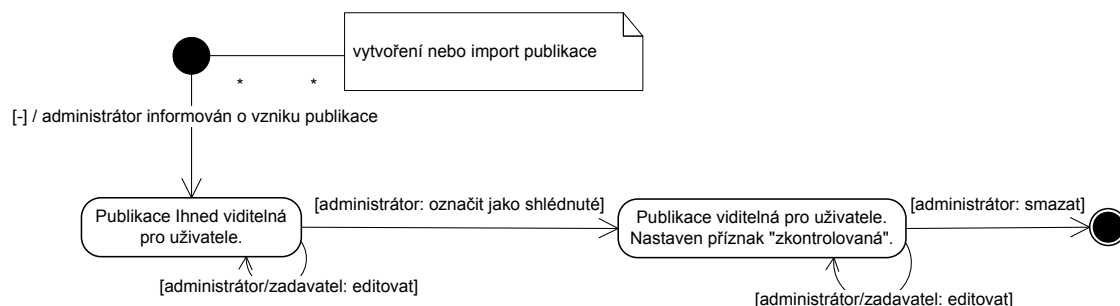
Potřeba autentizace a autorizačních mechanismů aplikace vyústila v požadavek vzniku úložiště dat o uživateli. Pro potřeby autentizace si vystačíme s tím, že uživatelům se bude přiřazovat přihlašovací jméno a heslo. Pro autorizaci uživatelů si vystačíme v databázi u každého záznamu s příznakem, který určuje roli, ve které bude uživatel vzhledem k systému figurovat. U uživatelů systému by bylo dále dobré evidovat jméno a email kvůli tomu, aby v případě ztráty hesla mohl uživatel požádat o nové, a tato žádost mohla být ověřena pomocí emailu.

## 2.3 Procesní požadavky

V této kapitole krátce popisují stavy, ve kterých se nacházejí některé objekty.

### *Publikace:*

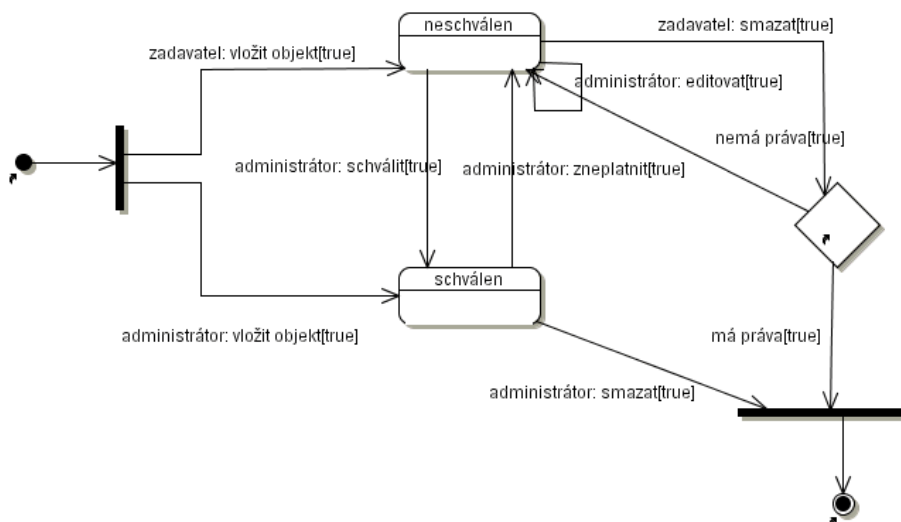
Následující diagram zdůrazňuje změnu (zjednodušení), kterou jsem provedl oproti původní implementaci, ve které publikace ještě než mohla být zobrazena, musela být schválena administrátorem. Schvalování bylo podle praktických zkušeností příliš restriktivní, proto bylo lépe ho odstranit a navrhnout mechanismus, který administrátorovi umožňuje snadno zjistit, které publikace byly přidány a chce-li, může je prohlédnout a poté označit jako „zkontrolované“.



**Obrázek 4 - Diagram stavů publikace**

### *Rozšiřující atributy:*

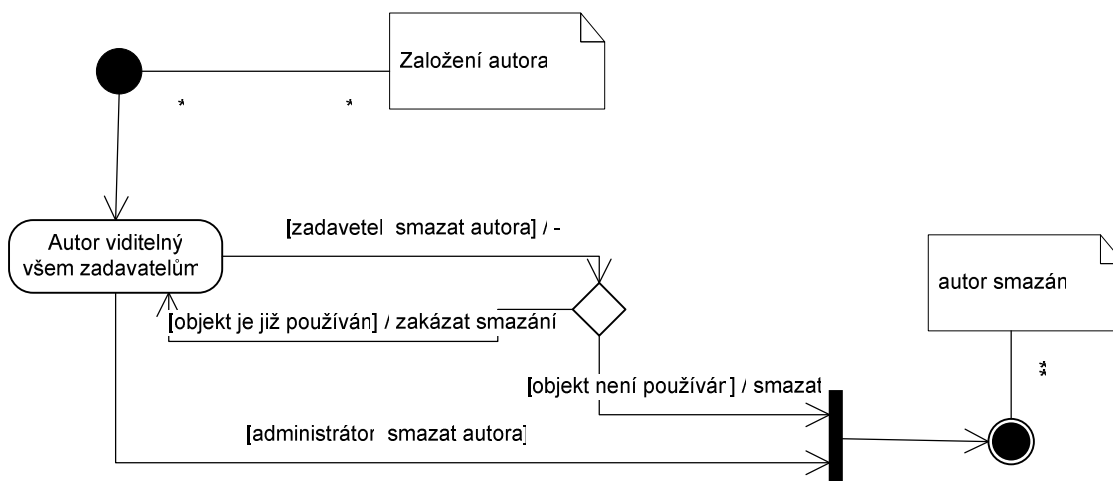
Následující diagram jsem převzal z původní dokumentace databáze publikací [1, kapitola 2]. Rozdíl však je v tom, že diagram se týká jen a pouze tzv. rozšiřujících atributů. Diagram vlastně popisuje mechanismus, kterým prochází tyto atributy – atribut může být vytvořen zadavatelem (i administrátorem). Administrátor může smazat i editovat jakýkoli atribut, ale zadavatel může pouze mazat atributy, které vytvořil jen on sám.



**Obrázek 5 - Stavový diagram objektů přidružených k publikaci**

#### *Autoři:*

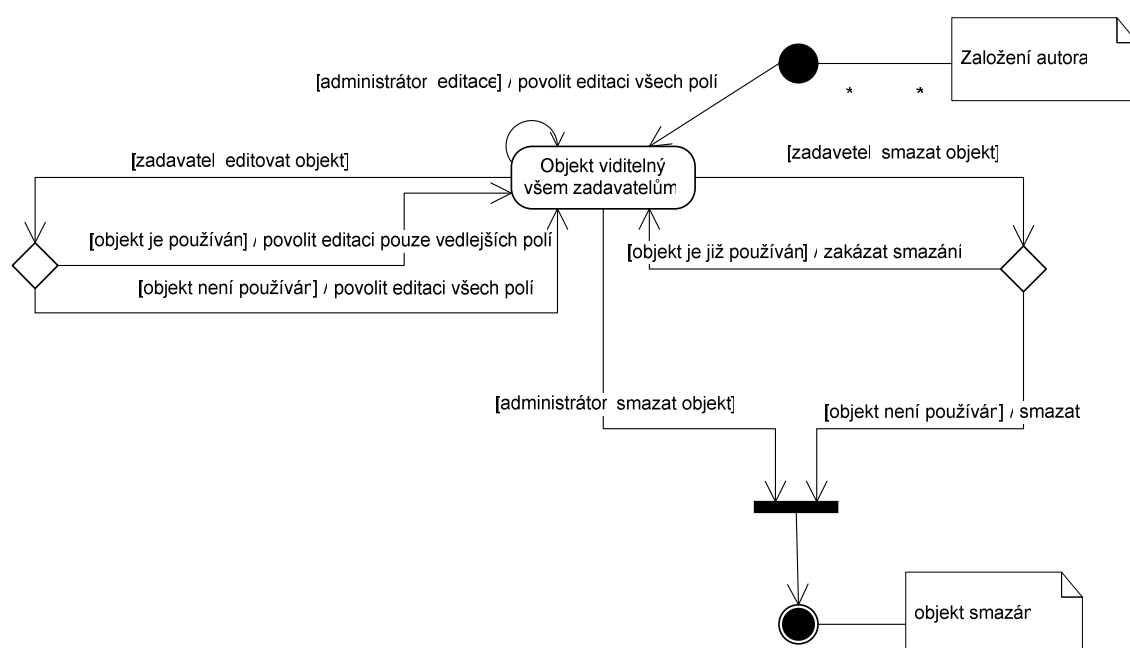
Životní cyklus objektu „autor“ je znázorněn na následujícím diagramu. Autor bude ihned po vytvoření přístupný k používání všem zadavatelům, ne jen uživateli, který záznam založil. Není potřeba, aby záznam s autorem schvaloval administrátor, jak tomu bylo dříve. Administrátor bude mít možnost smazat jakéhokoli autora s tím, že pokud na něj odkazují některé publikace, bude na tuto skutečnost pouze upozorněn. Zadavatel bude moci smazat pouze ty autory, kteří nejsou používáni publikacemi.



**Obrázek 6 - Diagram stavů objektu autor**

### Vydavatelé a časopisy:

Podobně jako autoři, tak i časopisy a vydavatelé, jsou objekty které se vážou na publikaci. V následujícím diagramu se snažím opět zdůraznit, že časopis a vydavatel po vytvoření nepodléhá procesu schvalování v administračním rozhraní. Co je však potřeba zdůraznit je, že jak zadavatel, tak administrátor mají mít možnost svým způsobem tyto záznamy modifikovat. Administrátor dostává neomezenou pravomoc, zatímco zadavatel je omezen. Totiž vydavatelé nebo časopisy, které byly již přiřazeny publikacím, nemůže mazat, a editovat je může jen v omezené míře. U vydavatele je vhodné zakázat editaci názvu, zatímco adresa není údajem, jehož neuvážlivou změnou by došlo k vážnému problému v konzistenci dat. Naopak se tím umožní běžným zadavatelům průběžně doplňovat chybějící údaje o vydavatelích. Podobná úvaha platí i pro časopisy. Zde je název časopisu údajem, který by se později, pokud je záznam časopisu používán, již neměl nikdy měnit.



Obrázek 7 - Stavový diagram časopisů a konferencí

### Konference:

U konferencí je situace podobná jako u časopisů a vydavatelů. Rozdíl je v tom, že unikátnost konference je určena názvem konference a rokem konání. S vedoucím práce jsme dospěli k závěru, že bude vhodné, aby zadavatel mohl vložit i neúplnou konferenci bez znalosti roku. V takovém případě by tato „neúplná konference“ neměla být k dispozici k výběru ostatním zadavatelům při vkládání nové publikace. Pokud na takovou konferenci neodkazuje žádná publikace, měl by se nějaký mechanismus postarat o to, aby byla tato konference smazána. Tento mechanismus by měl být aktivován, pokud byla konference přiřazena k publikaci, někdo tuto vazbu zrušil a počet publikací odkazujících na konferenci je roven nule.

## 2.4 Importy a exporty

V zadání práce je požadavek na import / export do standardních formátů (BibTeX, XML a protokol Z39.50).

### *BibTeX:*

Import a export z/do systému BibTeX má smysl, neboť spousta elektronických katalogů u svých publikací nabízí ke stažení spolu s PDF dokumenty také BibTeXové definice publikací. Import potom velice usnadní přebírání publikací. Stačí pouze zkopírovat definici a stáhnout dokument PDF a tyto vložit pomocí nějakého rozhraní do naší databáze. Export BibTeX definic se hodí např. při psaní LaTeX dokumentů.

### *XML:*

Export a import z/do XML v této práci implementovat nebudu, neboť jej v praxi nebude potřeba. Byl to jen jakýsi návrh na začátku, který se ale ukázal být naprosto zbytečný.

### *Z39.50:*

Z39.50 je velice rozsáhlý protokol určený především k vyhledávání záznamů v národních knihovnách. Protokol Z39.50 umožňuje paralelní vyhledávání a nabízí škálu vyhledávacích parametrů. Vznikl už v roce 1970. Od té doby se standard rozšířil o mnoho nových funkcí a bylo vytvořeno mnoho vzájemně téměř nekompatibilních implementací tohoto protokolu. V [16] je zkrácený seznam významných profilů protokolu Z39.50, podle kterých může být protokol konfigurován.

Když jsem se v průběhu analýzy zabýval protokolem Z39.50 zjistil jsem, že je velice komplikovaný a pro naši databázi publikací ne moc přínosný.

Nemá smysl implementovat export pomocí tohoto protokolu, neboť naše databáze publikací není určena veřejnosti. Import pomocí protokolu Z39.50 by měl smysl k přebírání záznamů ze světových knihoven. Zjistil jsem však, že import pomocí Z39.50 nebude mít pro nás takový význam, jak jsem původně očekával, a úsilí vynaložené na jeho implementaci bude veliké.

Dovolím si citovat [15] „Pokud se v současnosti rozhodneme využívat služeb Z39.50 serverů, zjistíme, že různé servery mají nejen odlišné vlastnosti, ale jejich chování je dokonce nezřídka v rozporu s protokolem Z39.50, takže je téměř nemožné vytvořit univerzálního Z39.50 klienta, který by dokázal komunikovat s libovolným Z39.50 serverem bez předchozí studie jeho chování“.

Vyzkoušel jsem několik klientů a webových bran založených na protokolu Z39.50. Za zmínku stojí Library of Congress Catalog [17] a Mercury Z39.50 client [18]. Výsledkem bylo to, že se mi potvrdil výrok konstatovaný v předchozím odstavci. Totiž že servery nevrací data jednotně. Dokonce nezřídka vracel server názvy autorů v poli kde klient očekává název publikace, datum konání conference, v adrese vydavatele apod.. Aby bylo možno realizovat smysluplně použitelný import, bylo by potřeba analyzovat chování každého konkrétního serveru.

Po prostudování možností a komplexnosti protokolu Z39.50 a zvážení jeho přínosů pro naši aplikaci jsme se s vedoucím práce rozhodli implementaci tohoto protokolu vyloučit a spíše se zaměřit na implementaci jiných, důležitějších rysů databáze publikací, které se objevily v průběhu analýzy i samotné implementace.



## 2.5 Použité technologie

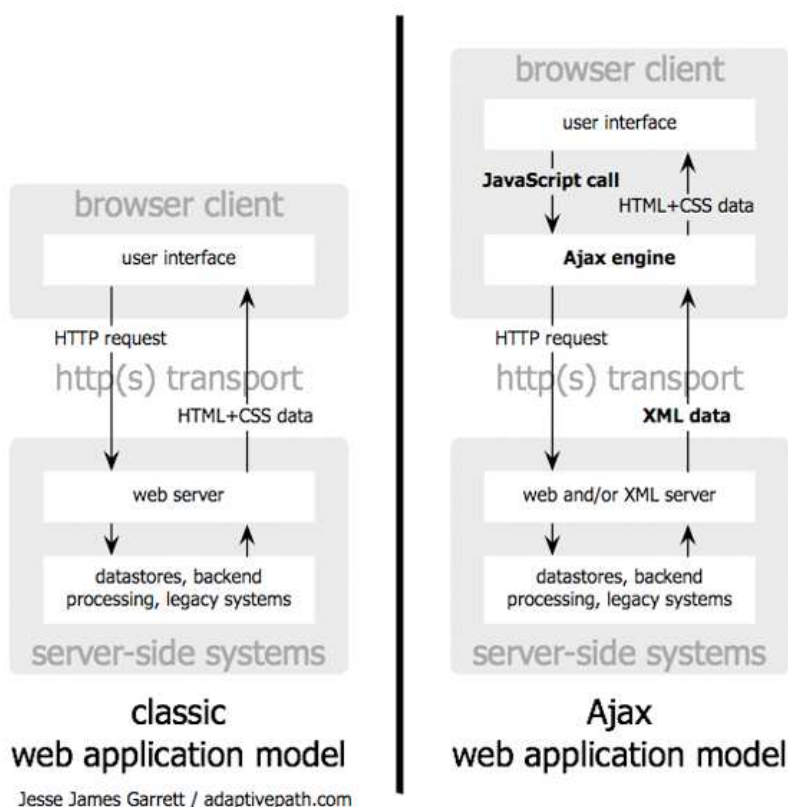
Volba implementačního prostředí byla dána v podstatě tím, že původní aplikaci jsem psal v PHP 5; že byla používána s databázovým strojem MySQL 5; přičemž pro přístup k databázi byl použit modul Pear DB [3]. Ačkoli jsem zpočátku váhal, zda by nebylo lépe přejít na jinou technologii než je PHP, po zvážení přínosů a ztrát jsem raději zůstal u PHP, neboť portace velkého množství kódu původní aplikace by bylo kontraproduktivní a výsledek tohoto snažení by nepřinesl velké klady, spíše jen problémy.

K implementaci některých částí aplikace jsem použil technologii Ajax, protože vylepšuje interaktivitu aplikace. Pro vyhledávání publikací je např. vhodné použít Ajaxový našeptávač, jak jej známe z některých vyhledávacích portálů. Technologii Ajax je vhodné použít i u formulářů na zadávání publikací. Formuláře byly už na začátku poměrně složité a těžko ovladatelné. Zadání této práce formuláře ještě zesložituje a já jsem proto hledal způsob jak zajistit, aby se v nich uživatel vyznal.

Myslím, že se sluší abych alespoň v krátkosti popsal co technologie Ajax je a k čemu se používá.

Termín AJAX se poprvé veřejně objevil r. 2005 v [13]. Slovo Ajax znamená „Asynchronous JavaScript and XML“. Ve zkratce, jde o techniku, při které se používá nestandardní objekt XMLHttpRequest, který komunikuje s aplikační logikou serveru. Může posílat i přijímat data v mnoha různých formátech, včetně XML a textu. Hlavním přínosem Ajaxu pro webové aplikace je možnost komunikovat „vpozadí“ se serverem a zobrazovat data na stránce bez jejího znovunačítání.

Na následujícím obrázku je porovnání „klasické“ a „Ajaxové“ webové aplikace. Diagram jsem převzal přímo od „autora pojmu Ajax“ z [13].



Obrázek 8 - Ajax: A New Approach to Web Applications

### 3 Zhodnocení existujících systémů

Myslím, že na tomto místě by neměly chybět systémy, které jsem popisoval v i mé první práci [1]. Pro úplnost zde tedy ve zkrácené podobě opakovaně prezentuji systémy BibAdmin, Bibloscape, EndNote a WriteNote. Nově zde popisuji systém Aigaion. Poslední skupinou aplikací, které nově prezentuji, jsou systémy CiteSeer, ACM Portal a RefWorks, které se svou velikostí nedají s mojí databází moc srovnávat, ale přesto je zde chci uvést, protože se mi líbí jejich myšlenka.

#### 3.1 Open Source systémy

##### 3.1.1 Aigaion

Aigaion [26] vznikl v roce 2004 jako projekt 4 členého týmu Phd studentů, kteří hledali způsob jak snadno organizovat reference na použité publikace během svého výzkumu. Proto vznikl tento projekt, který je šířen pod GPL licencí a jeho autoři ho udržují dodnes (leden 2009).

Systém je napsán v PHP a běží nad databázovým strojem MySQL.

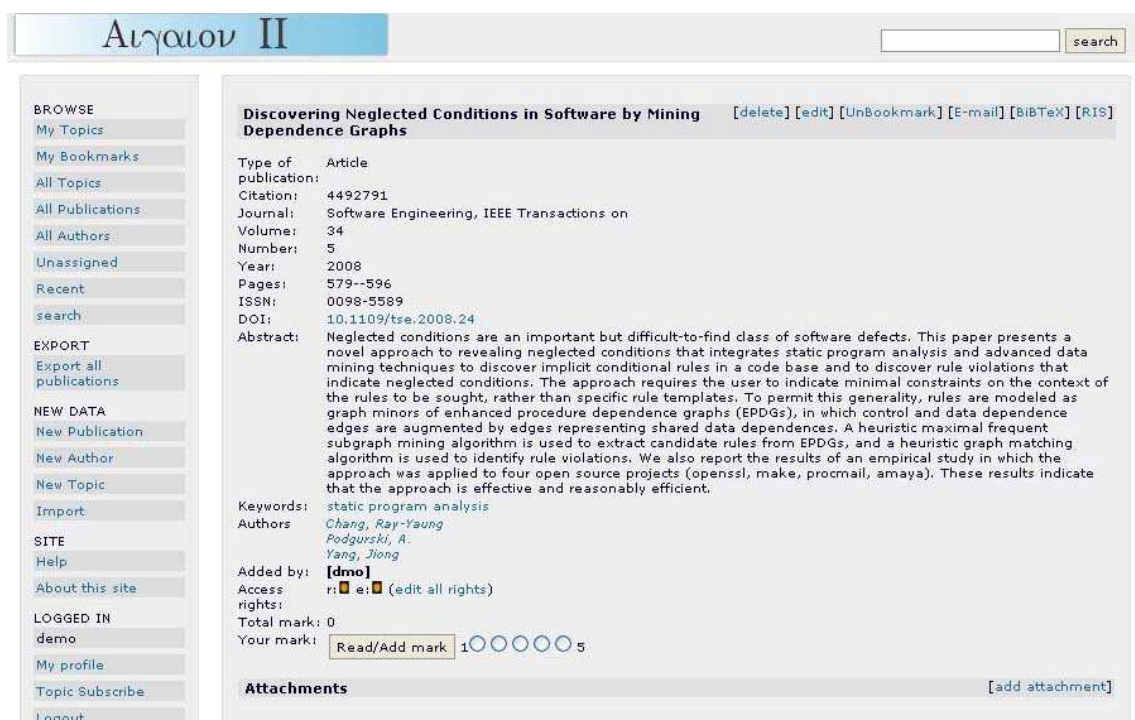
Hlavní rysy systému Aigaion:

- organizace publikací dle tématu
- možnost přiřazení záložek k publikacím
- možnost psaní poznámek k publikacím
- řazení publikací podle několika kritérií
- vkládání příloh k citacím
- vyhledávání podle různých kritérií
- import a export do formátů BibTeX a RIS
- formátování a export citací do html, nebo rtf
- nastavování individuálních uživatelských práv, od pouhých „čtenářů“ až po administrátory
- přiřazování uživatelů do skupin.

Publikacím lze přiřazovat příznaky o tom, zda mají být viditelné/editovatelné jen jejich autorem, interně nebo všemi uživateli. Uživatelům lze přiřazovat pomocí 12 příznaků konkrétní oprávnění jako např. právo čtení příloh, čtení poznámek editace příloh atd..

Aigaion neumí vyhledávat v samotném textu publikací. Vyhledává dle názvu, jmen autorů, uvnitř textu poznámek a abstractu. Umí filtrovat vyhledané výsledky podle tématu. Dále je schopen exportovat citace do formátů APA, British medical journal, Harward, IEEE, IEEE Trans., MLA a Turabian. Podporuje angličtinu, francouzštinu, holandštinu, němčinu a polštinu. Podpora jazyků však nefunguje spolehlivě.

Celkově se mi Aigaion jeví jako zdařilá aplikace.



Obrázek 9 - Vzhled webové aplikace Aigaion

### 3.1.2 Webový systém BibAdmin

BibAdmin [19] je webový správce bibliografických dat, skládá se z kolekce PHP skriptů, pracující s databází MySQL. Publikace se ukládají do databáze a mohou být přidávány, editovány a mazány prostřednictvím webového rozhraní. Je navržen hlavně pro výzkumné týmy, ke správě citací a publikací. Uživatelé mohou vkládat veřejné nebo soukromé bibliografie, které se pak dají použít ke kompilaci LaTeX dokumentů.

Vlastnosti:

- dva profily: první určen pro správu kompletních publikací a druhý ke správě citací; přístup uživatelů může být diferencován podle toho, zda jde pouze o citace nebo celé publikace
- vyhledávání podle klíčových slov
- upload souborů publikací typu PDF a PS a slidů (PDF, PPT)
- přidávání URL odkazů k citacím
- import/export BibTeXových definic
- licence GNU General Public Licence (GPL).

Po zběžném prohlédnutí tohoto systému musím konstatovat, že je v něm pár dobrých myšlenek. Např. podporuje formát bib, což se dobře uplatní při psaní LaTeX dokumentů. Citaci je možné stáhnout ve formátu bib. Zadávat publikaci jde buď přímo importem BibTeXových definic, a nebo pomocí formuláře, v němž jsou vyznačeny povinné a nepovinné položky. BibAdmin dělí uživatele na skupiny a na základě jejich oprávnění jim nabízí ke stažení pdf a jiné dokumenty. V administračním rozhraní lze přidělit těmto dokumentům rozsah viditelnosti, určující kteří uživatelé mohou dokument spatřit.

## 3.2 Komerční systémy

### 3.2.1 Biblioscape

Biblioscape [20] od firmy CG Information je software určený pro platformu MS Windows. Je součástí balíku programů, z nichž je dobré jmenovat ještě jeho zjednodušenou, neplacenou verzi BiblioExpress a webový server BiblioWeb. Biblioscape se používá k organizaci dokumentace, referencí a k získávání bibliografických dat na internetu. Biblioscape může být používán ke sbírání referencí na literaturu různých typů, k prohlížení bibliografických zdrojů na internetu a prohlížení citací v různých formátech.

Program jsem zkoušel a působil na mě dobrým dojmem, uživatelské rozhraní je rychlé a přehledné. Citace jsou rozděleny do složek, podle kterých se dělí např. do různých oborů, kategorií atd.. Zpravidla v pravé dolní části programu se zobrazuje výsledný formát citace daný některou ze zvolených norem. Dobrou funkcí programu, je možnost vytvářet odkazy na kompletní texty, ať už jde o URL odkazy nebo soubory uložené na lokálním počítači. Úplnou samozřejmostí jsou pokročilé metody vyhledávání, zálohování databází, kontrola duplicit. Vkládání nových citací je možné ručně nebo pomocí funkce „hot import“. Tato funkce umožňuje importovat citace přímo z webové stránky některého z katalogů (seznam katalogů je v programu zabudován).

Zaměřme se nyní na možnost, že pomocí programu BiblioWeb můžeme publikovat citace do prostředí internetu. BiblioWeb server generuje katalog publikací přístupný přes protokol http. Uživatelé do něj mohou nejen nahlížet, vyhledávat záznamy, ale mohou i do katalogu přidávat a editovat publikace. Na straně serveru stačí jen spustit soubor BiblioWeb.exe. Nastavení serveru je celkem intuitivní záležitost, webové rozhraní, které generuje, je uživatelsky přívětivé a částečně se blíží svojí funkcionalitou samotnému programu Biblioscape. V katalogu lze vyhledávat, filtrovat, zobrazovat publikace, editovat atd..

Co se týče typů publikací a jejich povinných a nepovinných údajů, Biblioscape si vede svůj vlastní standard, oproti BibTeXu rozděluje třeba typ article na journal article, magazine article a newspaper article. Přidává některé nové typy publikací, jako např. map, electronic source, film or broadcast, statute, hearing, artwork a další.

### 3.2.2 EndNote

Endnote [21] je určen pro platformu MS Windows. Je dílem společnosti Thomson ISI Researchsoft. EndNote je jedním z mnoha bibliografických produktů pocházejících z této firmy. K hlavním rysům patří např.:

- vkládání obrázků a jiných souborů k publikacím
- přístup k několika stům katalogů světových a univerzitních knihoven a bibliografickým databázím pomocí protokolu Z39.50
- 37 předdefinovaných typů citací (knihy, periodika, tabulky, CD apod.), možnost přidávat další
- pokročilé vyhledávání v knihovně citací s využíváním booleovských operátorů
- více než 1100 předdefinovaných stylů pro formátování citací, možnost tvorby vlastních
- Cite While You Write (vyhledávání a vkládání citací přímo z programu MS Word)
- možnost vystavení knihovny v prostředí internetu

- kontrola pravopisu, slovníky klíčových slov a synonym
- import citací z širokého spektra online i offline zdrojů.

### 3.2.3 WriteNote

WriteNote [22] je také dílem vývojářů z Thomson ISI Researchsoft. WriteNote je pouze webová služba, která umožňuje archivovat citace na vzdáleném serveru. Ke stažení je nabízen i WriteNote toolbar pro Microsoft Internet Explorer a MS Word. Má v sobě zabudovanou funkci „Cite While You Write“. Jde tak snadno udržovat konzistentní citace v dokumentech Wordu.

### 3.2.4 Webový portál CiteSeer

CiteSeer vznikl už v r. 1997 v institutu NEC Research, New Jersey; jeho zakladateli jsou Steve Lawrence, Lee Giles a Kurt Bollacker. V roce 2003 byl tento katalog přesunut na Pennsylvania State University's College of Information Sciences and Technology. Nedávno byla navržena nová architektura tohoto systému, kterou známe pod názvem CiteSeer<sup>x</sup> [23].

CiteSeer je veřejná univerzitní databáze publikací. Umožňuje skladování dokumentů PDF nebo PS, vyhledávání, řazení. Publikace se do této databáze zadávají jinak než u mojí databáze publikací. Katalog CiteSeer zahrnuje vyhledávacího robota, který prochází web. Nalezne-li nějaký dokument, rozloží jej na text a pomocí speciálního algoritmu zjistí informace o publikaci – jako např. název, autora, vydavatele, adresu autora, abstrakt a další. Dokáže také z dokumentů vyextrahovat citace a na základě nich odkazovat na ostatní dokumenty v databázi. Chcete-li zadat do systému CiteSeer svou publikaci, budete zpravidla požádáni o vložení URL, kde se vaše publikace nacházejí.

Algoritmus, který zadává informace o publikacích, nefunguje úplně vždy korektně. Ale i přesto se jedná o velmi zajímavý způsob zadávání publikací. CiteSeer je navržen tak, aby umožnil registrovaným uživatelům libovolně editovat publikace. Údaje o publikacích jsou verzované. Díky tomu může být CiteSeer veřejnou databází. Každý má možnost editovat cokoli, aniž by hrozilo riziko, že nějaký záškodník veškerá data smaže. CiteSeer dále umožňuje export BibTeX definic. K publikacím je možno přiřazovat štítky, podle kterých může registrovaný uživatel publikace zpětně vyhledat. Umožňuje zobrazení publikací, které jsou si podobné a mnoho dalších dobrých vlastností.

Jako negativum systému CiteSeer vidím pouze to, že dokumenty nejdou do databáze vložit ručně. Algoritmus automatické detekce vlastností publikace nefunguje (a ani nemůže) fungovat vždy korektně. Publikace v katalogu pak často obsahují neúplné nebo nekorektní údaje. Výhodou je, že díky verzování mohou uživatelé tyto údaje zpětně upravovat.

### 3.2.5 ACM Portal

ACM portal [24] je opět velmi rozsáhlý webový katalog publikací (především komerčních), který se nedá s mojí aplikací moc srovnávat. Uvádím jej zde spíše proto, že mě zaujalo jeho zpracování. Nabízí mnoho možností. Vyhledávání publikací, procházení sociální sítě autorů, zobrazování referencí mezi publikacemi, export do systémů BibTeX, EndNote a formátování referencí dle citačního formátu IEEE.

Dokumenty jsou ke stažení pouze zpravidla jen předplatitele.

### 3.2.6 RefWorks

RefWorks je komerční online systém pro správu referencí. Vznikl v roce 2001. RefWorks je služba, kterou neustále spravuje tým odborníků, který pružně reaguje na požadavky svých zákazníků a stále rozšiřuje schopnosti tohoto systému. Na stránce projektu [25] se můžete zaregistrovat a získat 30 denní testovací přístup.

RefWorks umožňuje uživatelům:


- vytvářet a spravovat osobní databázi referencí
- vyhledávat v osobní databázi podle různých kritérií
- import referencí z online zdrojů a různých formátů jako BibTeX, OneNote atd.
- formátování citací do více než 1800 různých citačních formátů
- export citací do formátu html, doc, rtf a odt
- sdílet reference s ostatními uživateli.

Zaměření systému RefWorks není, aby sloužil jako katalog publikací. Je především určen ke správě osobních referencí, a není možné jej využívat k ukládání samotných publikací.

Tento systém mě zaujal svojí uživatelskou přívětivostí a interaktivitou. Ač se jedná o webovou stránku, práce s ním připomíná práci s desktopovou aplikací. Pozoruhodný je také seznam zdrojů, ze kterých je možné reference importovat. Kromě samotného importu OneNote a BibTeXových definic nabízí dokonce možnost importu pomocí protokolu Z39.50. Import pomocí Z39.50 funguje kupodivu velmi spolehlivě.

Zde je nutno podotknout, že spolehlivosti importu z formátu Z39.50 je u systému RefWorks dosaženo tím, že pro každý Z39.50 server má RefWorks pevně předem dáno jak bude data ze serveru získávat. V kapitole 2.4 jsem se zmínil o tom, že není možné vytvořit univerzálního Z39.50 klienta. Tým, který vyvíjí systém RefWorks to řeší tak, že pokud chce uživatel používat nějaký Z39.50 server, který ještě není na výběr v systému RefWorks, pošle uživatel žádost s popisem serveru vývojářům. Ti později analyzují vlastnosti serveru a umožní uživatelům import i z tohoto nového serveru.

Za zmínku stojí, že RefWorks nabízí ke stažení nástroj Write-N-Cite, který pak může běžet v textovém procesoru a umožňuje pak efektivní práci při citování referencí v dokumentech word.


**RefWorks**

Welcome, Martin. [Log out](#)  
Individual User Subscription

References Search View Folders Bibliography Tools Help

Search RefWorks

**Import**
[Import Instructions](#) | [List of Filters & Databases](#) | [Request a Filter](#)

[Back to Reference List](#)

Import Filter/Data Source

Database

Import References into Folder

☒ **Import Data from the following Text File**

Encoding

☐ **Import Data from the following Text**

BibTeX

Los Alamos National Library  
CSA (Cambridge Scientific Abstracts)

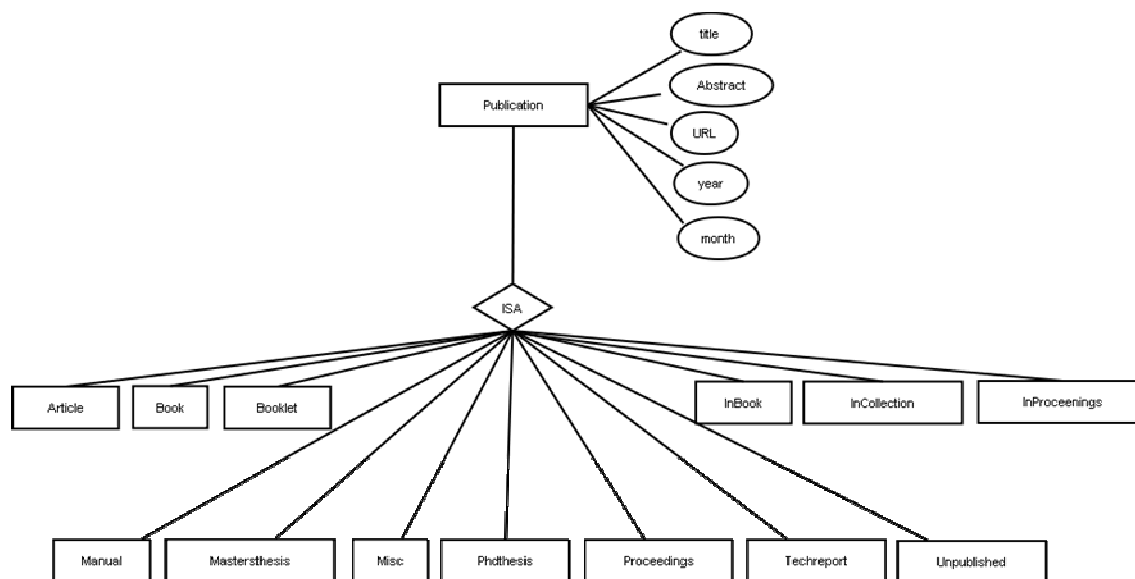
**Other**  
ABC-CLIO  
ABEKT - Full Format  
ACM Digital Library at eBridge  
ACS Publications  
Agris  
Albright College  
AMICUS - National Library of Canada  
Art and Archaeology Technical Abstracts  
Australian Literature  
AWI  
Bibliographie der deutschen Sprach- und Literaturwissenschaften - XML  
Bibliographix  
Bibliography of Asian Studies - BAS  
BiblioLine  
BIBSYS (RIS Import Format)  
BibTeX  
BIDS  
BOND Bibliothekssysteme

Obrázek 10 - Rozhraní webové aplikace RefWorks

## 4 Implementace

### 4.1 Datový model

Tím, že jsem začal dělit publikace do 13 typů místo původních čtyř, začal jsem uvažovat také o tom jak je vhodně promítnout do databáze.



Obrázek 11 - ER model hierarchie typů publikací

Předkem všech typů je entita publication, která má několik atributů společných pro všechny publikace viz obrázek 8. Její podtypy se liší množinami atributů, které podporují. Kompletní seznam atributů uvádím v příloze v kapitole 10.3.

Otázkou bylo, jak správně dekomponovat ISA hierarchii publikací do datového modelu. Jedna možnost je vytvořit pro každý typ publikace vlastní tabulku a z hlavní tabulky na ní odkazovat referencemi, přičemž atributy společné pro všechny typy publikací by byly umístěny v hlavní tabulce. Druhá možnost je, mít pro všechny publikace tabulku jednu s tím, že v ní bude vždy určitá množina atributů nevyužitá.

Výhoda první varianty je odstranění redundancí avšak za cenu složitějších SQL dotazů. Druhá varianta naopak nekomplikuje logiku aplikace, dotazy do databáze jsou jednodušší a rychlejší, avšak za cenu větší redundance.

Nakonec jsem se rozhodl zůstat u druhé varianty, protože tu využívala původní aplikace a myslím si, že větším problémem než redundance, je komplikovaná a nepřehledná aplikační logika.

Musím podotknout, že ani jeden z přístupů není dobrý. Cesta kterou jsem se vydal tiše toleruje redundance v databázi a těží z toho že větší část dat o publikaci je uložena v jedné tabulce. Druhý způsob by byl na druhou stranu čistší tím, že by se odstranily redundance, ale za cenu, že data publikace by se rozštěpila do dalších třinácti tabulek.

*Publikace typu článek, sborník a stať:*

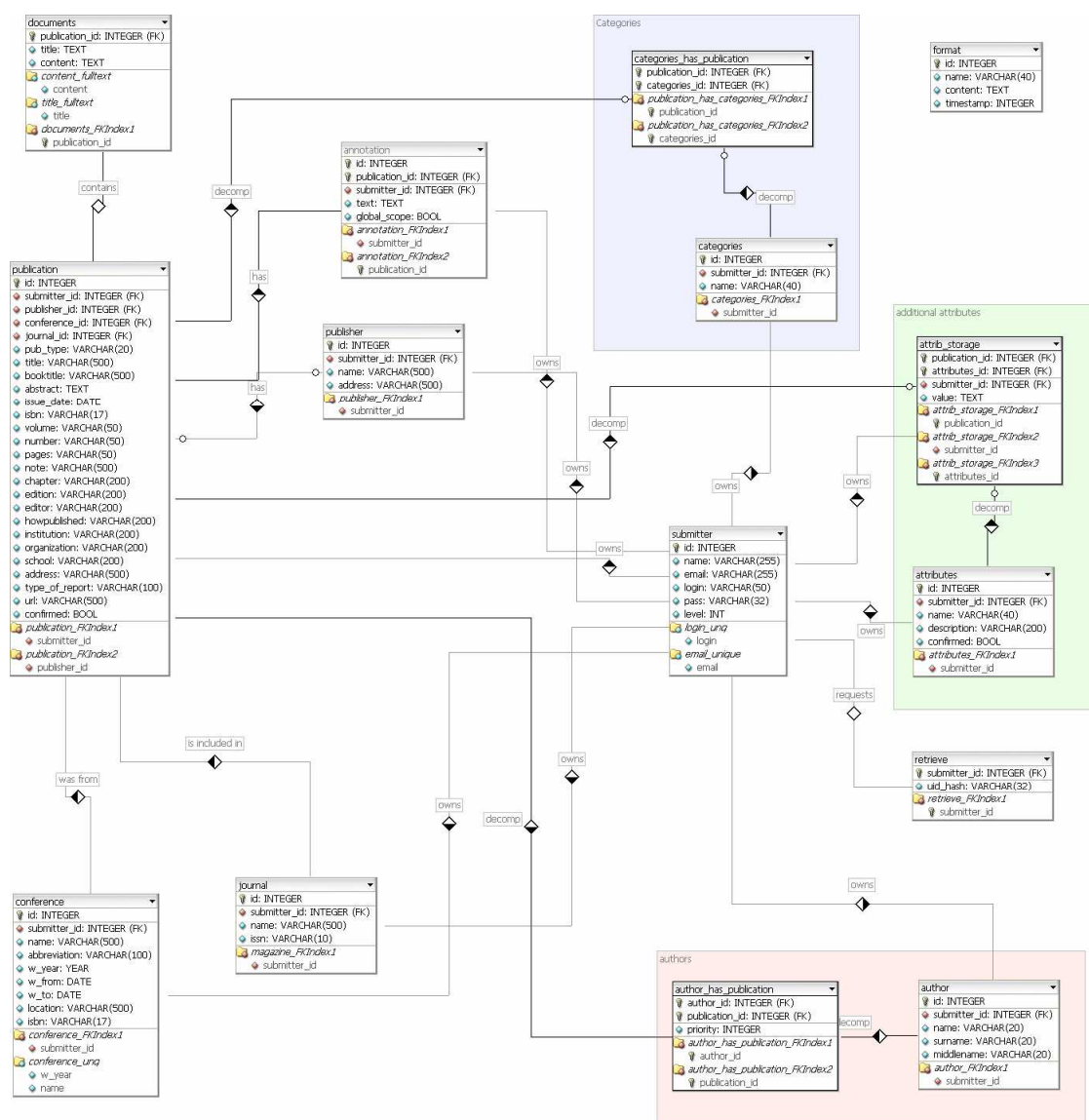
Je-li typem publikace sborník nebo stať ve sborníku z konference, bude záznam z hlavní tabulky odkazovat na pomocnou tabulku, která obsahuje seznam konferencí. Každá konference je identifikována jménem a datem, kdy se tato konference odehrávala. Výsledkem takové konference je sborník prací identifikovaný jednoznačným ISBN. ISBN je tedy vhodné také ukládat společně s údaji o konferenci do databáze. Tím vzniká určitá nejednoznačnost. Hlavní tabulka obsahuje sloupec ISBN, který se vhodně



použije, pokud typem publikace bude kniha. Problém nastává v okamžiku, kdy hlavní tabulka obsahuje publikaci typu proceedings nebo inproceedings. Tyto záznamy odkazují na tabulku conference, která také obsahuje sloupec ISBN. Této situaci se dá plně vyhnout pouze v případě, že bychom měli co typ publikace – to tabulka. Tento přístup jsem ale zamítl. Nezbyvá tedy nic než takovéto situace řešit na úrovni aplikační logiky.

Stejným způsobem budou odkazovat publikace typu „article“ na tabulku journal. Tabulka journal obsahuje ISSN a název časopisu nebo žurnálu, ve kterém článek vyšel.

Následující diagram znázorňuje datový model aplikace. Větší diagram naleznete na příloženém CD v adresáři documentation.



Pro cizí klíče jsou v tabulkách definovány indexy. Pro sloupce title a content v tabulce *documents* je definovány fulltext indexy. Pro tabulku *publication* jsem definoval update trigger, který synchronizuje sloupec title s tabulkou *documents*.

Seznam tabulek nebo úprav tabulek, které přibýly oproti původní implementaci:

*Tabulka publication:*

Obsahuje většinu vlastností publikace. Pouze údaje o vydavateli, autorech, konferenci nebo časopisu se ukládají do samostatných tabulek. Tabulka *publication* je s tabulkou *author* ve vztahu M:N, s tabulkou *publisher*, *conference* a *journal* ve vztahu 1:N.

Sloupce tabulky *publication*:

id (integer) ..... primární klíč  
submitter\_id (integer) ..... reference na tabulku submitter  
publisher\_id (integer) ..... reference na tabulku publisher  
conference\_id (integer) ..... reference na tabulku conference  
journal\_id (integer) ..... reference na tabulku journal  
pub\_type (integer) ..... typ publikace, nabývá hodnot v intervalu <1, 3>  
title (varchar) ..... název publikace  
booktitle (varchar) ..... název svazku  
abstract (text) ..... abstrakt  
issue\_date (date) ..... datum vydání publikace  
isbn (char) ..... ISBN  
volume (varchar) ..... volume  
number (varchar) ..... číslo  
pages (varchar) ..... počet stran nebo rozsah stran  
note (varchar) ..... poznámka  
chapter (varchar) ..... kapitola  
edition (varchar) ..... poznámky k vydání  
editor (varchar) ..... editor nebo editoři  
howpublished (varchar) ..... způsob publikování  
institution (varchar) ..... instituce  
organization (varchar) ..... organizace  
school (varchar) ..... škola  
address (varchar) ..... adresa  
type\_of\_reporr (varchar) ..... typ reportu  
url (varchar) ..... URL elektronického dokumentu  
confirmed (bool) ..... příznak, zda je publikace zkontrolována, či nikoliv

*Tabulka journal:*

Slouží jako číselník časopisů. Odkazují na ní záznamy z tabulky *publication* typu „article“, u kterých víme, ze kterého časopisu byly převzaty. Ukládá se do ní název časopisu a ISSN.

Sloupce tabulky *journal*:

id (integer) ..... klíč  
submitter\_id (integer) ..... reference do tabulky submitter  
name (varchar) ..... název časopisu  
issn (char) ..... ISSN

#### *Tabulka conference:*

Slouží jako číselník konferencí. Odkazují na ní záznamy z tabulky *publication* typu „inproceedings“ a „proceedings“. Ukládá se do ní název konference, zkratka názvu konference, místo konání, rok, datum začátku a konce konference, ISBN sborníku.

#### *Sloupce tabulky conference:*

id (integer) ..... klíč  
submitter\_id (integer) ..... reference do tabulky submitter  
name (varchar) ..... název konference  
abbreviation (varchar) ..... zkratka názvu  
w\_year (year) ..... rok konání  
w\_from (date) ..... datum zahájení konference  
w\_to (date) ..... datum ukončení konference  
location (varchar) ..... místo konání  
isbn (char) ..... ISBN sborníku

#### *Tabulka publisher:*

Původní tabulku „publisher“ jsem rozšířil o sloupec „address“. Uchovává adresu vydavatele. Odstranil jsem sloupec „confirmed“, který sloužil jako příznak schválení.

#### *Sloupce tabulky publisher:*

id (integer) ..... klíč  
submitter\_id (integer) ..... reference do tabulky submitter  
name (varchar) ..... název vydavatele  
address (varchar) ..... adresa vydavatele

#### *Tabulka annotation:*

Slouží k ukládání poznámek k článkům. Ukládá se do ní text poznámky a příznak o tom, zda je poznámka globální.

#### *Sloupce tabulky annotation:*

id (integer) ..... klíč  
publication\_id (integer) ..... reference do tabulky publication  
submitter\_id (integer) ..... reference do tabulky submitter  
text (varchar) ..... text poznámky  
global\_scope (bool) ..... příznak globálnosti

#### *Tabulka documents:*

Tato tabulka vznikla speciálně pro potřeby MySQL fulltextového vyhledávání. Fulltextové vyhledávání v MySQL potřebuje pro svoji funkci fulltextové indexy, které je možné vytvořit pouze u tabulek typu MyISAM, zatímco ostatní tabulky v databázi jsou typu InnoDB a jsou vhodné pro transakční zpracování. Problematice fulltextového vyhledávání se věnuji v kapitole 4.3.1.

#### *Sloupce tabulky document:*

publication\_id (integer) ..... reference do tabulky publication - je zároveň klíčem

title (text) ..... název publikace

content (text) ..... text publikace převzatý z pdf/ps dokumentů

#### *Tabulka submitter:*

Původní tabulku *submitter* jsem rozšířil o sloupce „name“, „email“, „pass“ a „level“.

#### *Sloupce tabulky submitter:*

id (integer) ..... klíč

login (varchar) ..... přihlašovací jméno

pass (char) ..... hash přihlašovacího hesla

name (varchar) ..... jméno uživatele

email (varchar) ..... email uživatele pro potřeby obnovení hesla

level (integer) ..... úroveň oprávnění (0 administrátor, 1 zadavatel, 2 uživatel)

#### *Tabulka retrieve:*

Tabulku *retrieve* jsem založil za účelem uchování dočasných dat při procesu obnovení ztraceného hesla. Proces obnovy hesla popisují v kapitole 4.2.

#### *Sloupce tabulky retrieve:*

submitter\_id (integer) ..... reference do tabulky submitter – je zároveň prim. klíčem

uid\_hash (char) ..... hash uid

#### *Tabulka categories:*

Z tabulky *categories* jsem odebral sloupec *confirmed*, neboť kategorie již nebudou procházet procesem schvalování. Jsou do ní ukládány názvy kategorií, do kterých jednotlivé publikace patří. S entitou *publication* je ve vztahu m:n, tento vztah je dekomponován na úrovni tabulky *categories\_has\_publication*

#### *Sloupce tabulky categories:*

id (integer) ..... klíč

submitter\_id (integer) ..... reference do tabulky submitter

name (varchar) ..... název kategorie

#### *Tabulka categories\_has\_publication:*

Plní funkci dekompoziční entity (upravuje relaci m:n mezi tabulkami publication a categories na dvě relace typu 1:n). Zůstává beze změny.

Sloupce tabulky *categories\_has\_publication*:

publication\_id (integer) ..... reference do tabulky publication, je součástí klíče

categories\_id (integer) ..... reference do tabulky categories, je součástí klíče

#### *Tabulka author\_has\_publication:*

Je dekompoziční tabulka, jenž svazuje řádky tabulky publication a tabulky autor. Přidal jsem do ní sloupec „priority“. Pomocí něj se dá určit pořadí autorů podle významnosti.

Sloupce tabulky *author\_has\_publication*:

author\_id (integer) ..... reference do tabulky author, je součástí prim. klíče

publication\_id (integer) ..... reference do tabulky publication, je součástí prim. klíče

priority (integer) ..... priorita autora

#### *Tabulka attributes:*

Zůstává oproti původní implementaci beze změny. Je určena k evidenci rozšířených atributů k publikacím. S entitou publication je ve vztahu m:n, tento vztah je dekomponován na úrovni tabulky attrib\_storage.

Sloupce tabulky *attributes*:

id (integer) ..... primární klíč

submitter\_id (integer) ..... reference do tabulky submitter

name (char) ..... jméno atributu

description (char) ..... popis atributu

confirmed (bool) ..... příznak, zda je záznam o atributu veřejný, či nikoliv

#### *Tabulka attrib\_storage:*

Zůstává oproti původní implementaci beze změny. Plní funkci dekompoziční entity (upravuje relaci m:n mezi tabulkami publication a attributes na dvě relace typu 1:n). Navíc tato tabulka obsahuje úložiště konkrétních atributů, které náleží jednotlivým publikacím.

Sloupce tabulky *attrib\_storage*:

publication\_id (integer) ..... reference do tabulky publication, je součástí prim. klíče

attributes\_id (integer) ..... reference do tabulky attributes, je součástí prim. klíče

submitter\_id (integer) ..... reference do tabulky submitter

value (text) ..... hodnota atributu

#### *Tabulka format:*

Zůstala beze změny. Je to tabulka, která není v relaci s ostatními tabulkami. Slouží

k ukládání Smarty šablon určených k formátování citací.

Sloupce tabulky *format*:

id (integer) ..... klíč

name (char) ..... název normy, podle které je šablona sestavena

content (text) ..... text Smarty šablony

timestamp (integer) ..... časové razítko šablony

## 4.2 Autentizace a autorizace

Databázi publikací jsem rozšířil o možnost autentizace zadavatelů i čtenářů, což bylo dříve možné jen v omezené míře. Předpokládá se, že aplikace **není** veřejně přístupná. Dokumenty v ní uložené, jsou chráněny autorským zákonem a nemají být nabízeny libovolně ke stažení. Proto je potřeba vyžadovat přihlašovací údaje po všech uživateli, kteří databázi používají, tedy i těch, kteří v databázi jen vyhledávají publikace. Při vstupu do aplikace se vždy zobrazí autentizační formulář, kam uživatel vloží své jméno a heslo. Na základě přihlašovacích údajů se konkrétnímu uživateli přiřadí jeho role. Publikace, které zadavatel vloží do databáze, jsou opatřeny jeho identifikátorem a lze zpětně dohledat, kdo určitou publikaci vkládal nebo modifikoval. Pokud některý uživatel modifikuje publikaci, kterou vytvořil jiný uživatel, bude možno zpětně dohledat jen toho, kdo publikaci modifikoval a nikoliv původního autora záznamu.

Původní aplikace používala HTTP autentizaci verze 1.0. Tento protokol však neumožňoval bezpečné předání hesla. Heslo se odesílá v otevřené formě po síti, a to při každém odeslání klientského požadavku na server. Nový autentizační mechanismus, který jsem implementoval, používá k předání hesel protokol HTTPS.

Samotnou autentizaci má na starosti třída *AuthModule* (/common/AuthModule.inc.php). Objekt této třídy udržuje po přihlášení session, s číslem aktuálně přihlášeného uživatele, a zpřístupňuje ostatním programovým modulům metody pro :

- přihlášení/odhlášení uživatele
- změnu hesla
- obnovení ztraceného hesla
- vytvoření/smazání uživatele
- zjištění údajů o aktuálním uživateli (jméno, číslo a role)
- nastavení jiného aktuálního uživatele

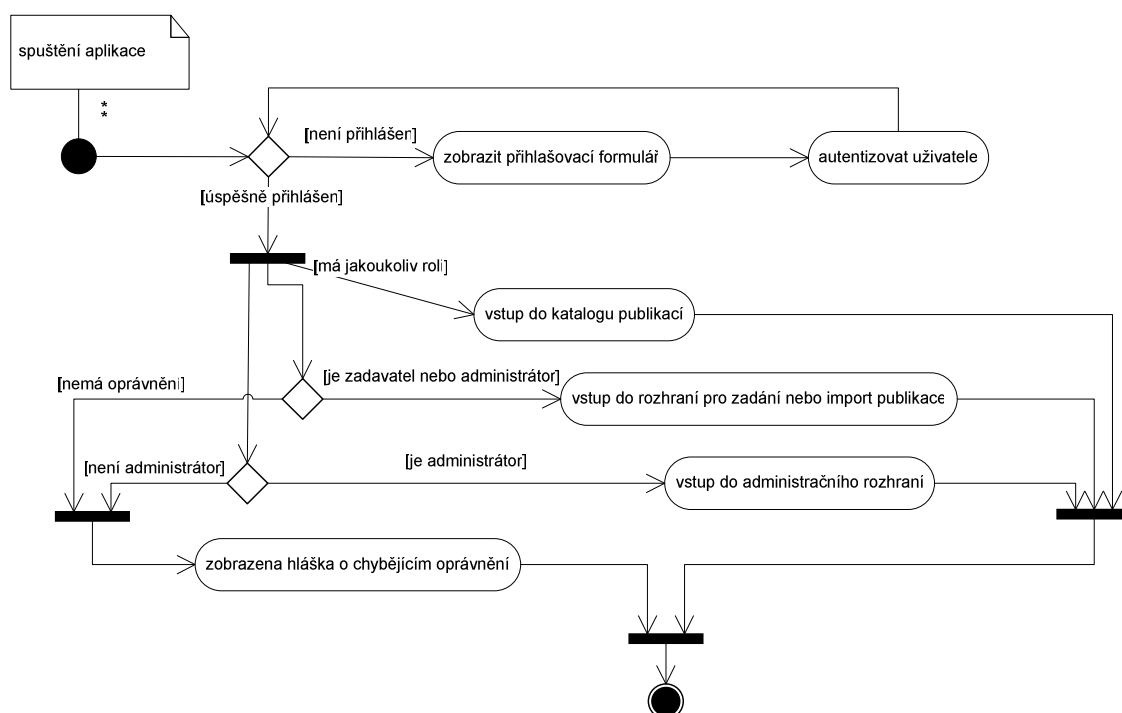
Data o uživateli si aplikace ukládá do tabulky *submitter*. Místo hesel se z bezpečnostních důvodů do databáze ukládají pouze md5 otisky hesel, která se při přihlašování uživatele kontrolují s otiskem hesla zadaných do přihlašovacího formuláře.

*Přihlašovací formulář a uživatelský účet:*

Logiku formuláře implementuje třída *Account* (v souboru /account/Account.inc.php). Vzhled formuláře je definován v šabloně /templates/account.tpl.

Otevře-li nepřihlášený uživatel některý z modulů databáze publikací, dojde k přesměrování na zabezpečený přihlašovací formulář. Po úspěšném přihlášení je zpět

přesměrován na konkrétní modul, který chce použít viz následující obrázek:

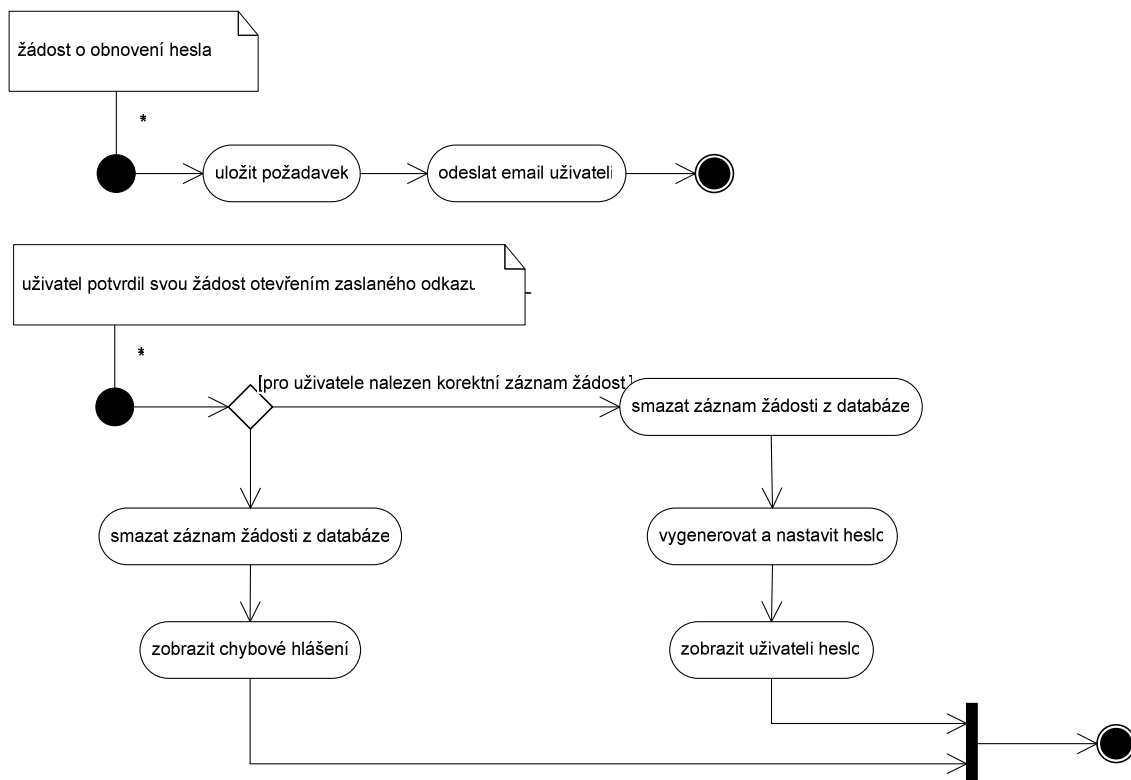


**Obrázek 12 - Přihlášení uživatele**

### *Obnovení ztraceného hesla:*

Funkci obnovení ztraceného hesla lze vyvolat z formuláře pro přihlášení. O zpracování požadavku na obnovení ztraceného hesla se stará opět třída *AuthModule*. Při tomto procesu se používá tabulka *retrieve*, kam se ukládá hash unikátního identifikátoru. Tento unikátní identifikátor je vygenerován při každém požadavku na obnovu hesla, je zaslán spolu s informační zprávou na email daného uživatele a jeho md5 otisk se uloží do databáze.

V první fázi dojde vlastně k autentizaci tím, že se zpráva odešle na email správného příjemce, a ten svou vůli obnovit heslo vyjádří tím, že klikne na odkaz s unikátním identifikátorem. Shoduje-li se otisk tohoto identifikátoru s tím který byl předtím vložen do databáze, vygeneruje autentizační modul nové heslo a zobrazí ho uživateli. Diagram činnosti obnovení hesla je na následujícím obrázku:



Obrázek 13 - Obnovení hesla

### 4.3 Implementace katalogu publikací

Aplikační logika katalogu publikací se nachází ve třídě *User* (v adresářové struktuře pod `/User.inc.php`). Soubor stylů je v souboru `/user.css`. Vzhled uživatelského rozhraní je uložen v šabloně `user.tpl` a `user_publ.tpl` (detail publikace).

Původní implementace fungovala tak, že na hlavní straně se okamžitě zobrazovaly publikace, aniž byla předtím zadána vyhledávací kritéria. Od tohoto způsobu zobrazování jsem upustil a nechal hlavní stranu prázdnou.

Užitečnou změnou, kterou jsem provedl, je úprava vzhledu vyhledávacího formuláře. Praxe ukázala, že není příliš vhodné, aby byl vyhledávací formulář složitý a bohatý na ovládací prvky. Tato složitost se odrazila v malé intuitivnosti používání vyhledávání. Na místo starého složitého formuláře jsem navrhl formulář se dvěma záložkami. První záložka slouží k fulltextovému vyhledávání v dokumentech a druhá záložka slouží k vyhledávání autorů.

Vyhledávání dle kategorií se ukázalo být stále vhodné, a proto jsem jej v této nové verzi také použil. Pouze jsem ho doplnil o operátory AND a OR, lze s nimi specifikovat zda vyhledávané publikace mají patřit do všech vybraných kategorií nebo alespoň do jedné z nich.

Uživatel má k dispozici pouze textové pole a tlačítko odeslat. Dále má možnost vybrat do kterých kategorií mají publikace patřit. Nechávám uživateli zobrazit vyhledávací pole, kam zadá buď úryvek hledaného textu, nebo jména autorů. Má na výběr, zda bude výsledek hledání filtrovat podle kategorií a zda mají nalezené publikace obsahovat všechny vybrané kategorie nebo alespoň některé z nich.

Nevyplní-li uživatel úryvky textu ani jména autorů, ale vybere-li některé kategorie



z nabídky, budou nalezeny publikace, které odpovídají vybraným kategoriím.

Vstupním bodem aplikace je skript /index.php. V něm se vytváří instance třídy *User*, která dostává jako parametry konstruktoru pole POST a GET ošetřená na výskyt speciálních znaků (jednoduchá uvozovka, dvojité uvozovky, backslash). Vyhledávací kritéria se předávají výhradně pomocí metody GET.

Zde uvádím seznam parametrů, které se předávají prostřednictvím metody GET:

Keywords.....fráze hledaná v textu publikací

Authors .....jména autorů

Categories[] .....pole id kategorií podle kterých se má filtrovat

Operator .....{and,or} příznak zda musí být ve všech kategoriích

Page .....číslo stránky výsledku

Order\_by .....{title, date, relevancy}, řazení výsledků

Advanced .....identifikuje, zda má zůstat aktivní vyhledávání dle kategorií

#### 4.3.1 Fulltextové vyhledávání

Vloží-li zadavatel do databáze publikací, získá se automaticky z příložených pdf/ps souborů text, který se uloží do tabulky *documents*. Text získaný z příložených pdf/ps souborů se poté využije pro fulltextové vyhledávání pomocí standardních funkcí databázového stroje MySQL [4].

Pro extrakci textu z dokumentů PDF/PS používám externí programy, které se volají z PHP skriptů aplikace. Programovým modulem, který má na starosti extrakci textu z dokumentů je třída *Convertor* ze souboru /common/Convertor.inc.php. Detailněji se této problematice věnuji v kapitole 4.4.

Fulltextová data ukládám do tabulky *documents*, která je svázána s tabulkou *publication* v relaci 1:1. Pokud tedy je k publikaci přiloženo více dokumentů pdf/ps, ukládám jejich fulltextová data společně do jednoho textového sloupce v tabulce *documents*.

Tabulku *documents* jsem zavedl v datovém modelu, protože fulltextové vyhledávání, které má v sobě databáze MySQL, může fungovat pouze nad tabulkami MyISAM, které ale nepodporují transakční zpracování jako ostatní tabulky, které používám k ukládání ostatních dat. V tabulce *documents* ukládám pouze data potřebná pro fulltextové vyhledávání (název publikace a text publikace). Ostatní data zůstávají v tabulce *publication*.

Vyhledávání probíhá v BOOLEAN módu [4, kapitola 11.8.2]. Prohledává se nejen text publikací, ale i samotné názvy publikací. Výsledek hledání řadím dle relevance vyhledávání v přirozeném jazyce, přičemž upřednostňuji publikace, u nichž byla klíčová slova nalezena i v názvu.

Řadit lze i podle názvu nebo roku vydání publikace. Nechávám uživateli možnost, aby výsledky seřadil dle libosti.

Seznam nalezených publikací zobrazuji na hlavní stránce společně s dalšími informacemi o publikacích, kterými jsou:

- odkaz na detail publikace

- autoři publikace
- kategorie, do kterých publikace patří
- úryvky nalezeného textu

Jelikož MySQL sama od sebe úryvky nalezeného textu nevrací, generuji úryvky textu programově. Nalezená klíčová slova v úryvcích zvýrazňuji tučným písmem.

#### 4.3.2 Vyhledávání podle jmen autorů

Webovou databázi publikací jsem rozšířil o vyhledávání dle jmen autorů. Původní databáze publikací tuto možnost měla velice omezenou. Vyhledávání podle autorů bylo předtím realizováno pouze jen jako filtr, který omezil výslednou množinu nalezených výsledků podle jednoho autora, který byl vybrán v menu. Tento způsob byl jednoduchý, ale ne příliš flexibilní, a v praxi se ukázalo že vhodnějším řešením by bylo vyhledávání autorů jiným, lepším způsobem. Navrhl jsem tedy jiný způsob vyhledávání:

Uživatel bude mít možnost do vyhledávacího textového pole zapsat libovolný počet jmen autorů a nalezený výsledek bude obsahovat publikace, u nichž se vyskytuje jako autor alespoň jeden z vyhledávaných.

Vyhledávání podle autorů probíhá tak, že se napřed hledaný řetězec rozdělí na slova, a každé slovo se pak porovnává se jménem, prostředním jménem a příjmením konkrétního autora. Vzhledem k tomu, že v tabulce *author* jsou uložena jen jména autorů a ne rozsáhlý text, vyhledávám záznamy pomocí klauzule LIKE.

Abych vyhledávání dle autorů usnadnil, zprovoznil jsem k tomuto účelu našeptávač podobný těm, které používají některé webové vyhledávače. O technologii Ajax jsem se již zmínil v závěru analýzy. Pro implementaci našeptávače jsem použil Ajax knihovnu AjaxAC 0.4.5 [5]. Jde o open-source white box framework napsaný v jazyce JavaScript a PHP, vydaný pod licencí Apache 2.0.

Moje implementace se nachází ve třídě *Suggest* v souboru */Suggest.inc.php* a rozšiřuje třídu *AjaxACApplication*. Uživatel, který do vstupního pole zadává text, ihned vidí seznam autorů jejichž jméno, prostřední jméno nebo příjmení začíná prefixem, který uživatel zadal.

Řadit výsledky vyhledávání lze podle názvu nebo roku vydání publikace. Seznam nalezených publikací zobrazuji na hlavní stránce společně s dalšími informacemi o publikacích, kterými jsou:

- odkaz na detail publikace
- autoři publikace
- kategorie, do kterých publikace patří.

#### 4.3.3 Zobrazení detailu publikace:

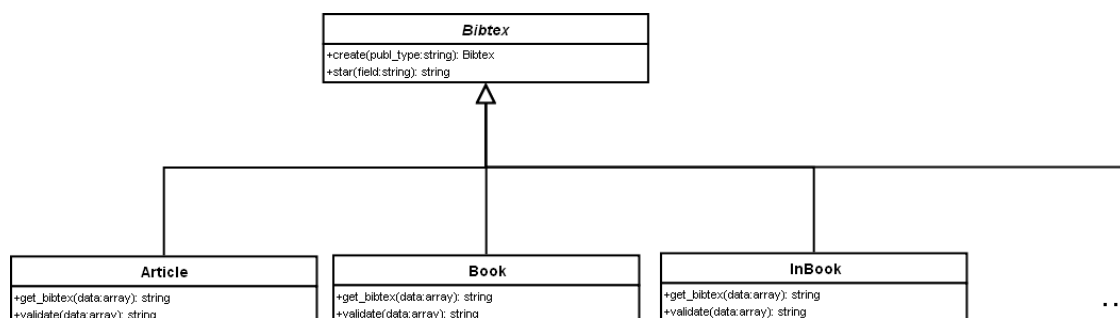
Databázi publikací bylo nutno rozšířit o možnost zobrazení detailu publikace. Původní implementace zobrazovala pouze seznam publikací naformátovaných v podobě normovaných citací, přičemž ke každé citaci se zobrazoval odkaz na přiložené dokumenty.

Detail publikace obsahuje nyní více podrobností k publikaci, BibTeX definici publikace, přehledný seznam dokumentů, citaci s možností volby jejího formátu a dále záložky určené k modifikaci publikace a psaní poznámek. Záložka pro modifikaci publikace (přístupná pouze zadavatelům a administrátorům) nabízí stejné možnosti jako 1. záložka administračního rozhraní. Zadavatel v ní může libovolně modifikovat vlastnosti publikace, zakázáno je však mazání publikace. Třetí záložka je určena k přidávání nebo modifikaci poznámek.

#### *Export publikací do BibTeXu:*

K tomuto účelu jsem navrhl třídu *Bibtex* a její podtřídy (všechny se nacházejí v souboru `/common/Bibtex.inc.php`). Jejich účel není jen generování definic BibTeXu, ale i validace BibTeXových polí při zadávání/importu publikací a generování značek, které vyjadřují povinnost ve formuláři zadávání publikací.

Hlavní třídu *Bibtex* rozšiřují její potomci – třídy *Article*, *Book*, *Booklet*, *InBook*, *InProceedings* atd.. Konkrétní instance těchto tříd lze získat pomocí statické metody *create* třídy *Bibtex*, která plní funkci factory. Nad každou instancí lze volat metody *validate*, kde parametrem je asociativní pole obsahující BibTeXová políčka; metoda *star*, kde parametrem je název BibTeXového políčka a výsledkem je příznak o povinnosti políčka podle typu publikace. Dále je to metoda *get\_bibtex*, která vrátí jako výsledek BibTeXovou definici na základě parametru ``data``, který byl této metodě předán.



**Obrázek 14 - Diagram tříd BibTeX**

Každá podtřída třídy BibTeX ví, která pole jsou pro určitý typ publikace povinná a která volitelná. Přehledná tabulka s BibTeXovými poli se nachází v příloze v kapitole 10.3.

#### *Stahování dokumentů:*

Publikace může mít k sobě přiloženo libovolné množství dokumentů. Zpravidla se jedná o soubory PDF, ale mohou to být i soubory PS a další. Seznam povolených souborů je definován v souboru `/common/common.inc.php`. Jako úložiště souborů používám (stejně jak je tomu u původní verze) souborový systém. Soubory se ukládají do adresáře `/storage/[číslo publikace]/`.

Vzhledem k tomu, že databáze publikací nemá být veřejně přístupná, musel jsem vyřešit ještě otázku toho, jak zabezpečit dokumenty proti odcizení osobou, která nemá mít k databázi publikací přístup. To, že se dokumenty ukládají do souborového úložiště, by

mohlo útočníkovi umožnit jejich „vykradení“. Tuto otázku jsem vyřešil tak, že jsem zakázal HTTP přístup do složky /storage/ souborem `.htaccess`. Stahování souborů je nyní možné jen přes php skript, který nejprve ověřuje identitu uživatele a teprve potom odesílá uživateli obsah vyžádaného souboru.

Skript, který odeslání souboru zprostředkovává je `/document_provider.php`. Tento skript používá třídu *DocumentProvider* (v souboru `/common/DocumentProvider.inc.php`). Třída *DocumentProvider* kromě zmíněné funkčnosti nabízí ještě metodu, která na základě předaného parametru – id publikace vrací seznam souborů, které jsou k publikaci přiloženy.

Souboru `document_provider.php` je potřeba předat následující parametry metodou GET:

pid .....id publikace

f .....jméno souboru

*Psaní poznámek k publikacím:*

#### 4.3.4 Generování citací podle normy ČSN a IEEE

Implementoval jsem export citací dle normy ČSN ISO 690 podle [6] a [7] a dle normy IEEE [8]. Výsledkem jsou šablony, které jsem uložil do tabulky format pod názvem ČSN ISO 690 a IEEE. Co se týče obou norem, shrnu hlavní vlastnosti těchto stylů v bodech a uvedu příklady ve formě, jak jsem je převzal z [7] a [8].

*Styl ČSN ISO 690:*

Struktura citace:

Kniha nebo část knihy:

Primární odpovědnost, “Název díla nebo kapitoly”, in název svazku, Označení vydání. Místo vydání: Jméno nakladatele, Rok vydání. Rozsah díla. Edice. Poznámky. Standardní číslo. Lokace ve zdrojovém dokumentu, URL: <url dokumentu>

Příspěvek ve sborníku:

Autoři. Název příspěvku. In Název zdrojového dokumentu. Primární odpovědnost za zdrojový dokument. Vydání. Místo vydání: Jméno nakladatele, Rok. Lokace ve zdrojovém dokumentu.

Článek:

Odpovědnost za příspěvek. Název příspěvku. Sekundární odpovědnost. Název zdrojového dokumentu, Rok, ročník, číslo svazku, lokace části. Standardní číslo - ISSN.

Pozn. Podtržené položky jsou povinné stejně jako psaní názvu kurzívou.

**- Autorská odpovědnost** se uvádí ve formě:

*Příjmení, křestní jméno nebo iniciály křestního jména*

Zpravidla se příjmení autora uvádí velkými písmeny. Jméno se uvádí bez titulu. Je-li v dokumentu uvedeno více autorů, uvádíme v takovém případě všechny autory do počtu tří. Nad počet tří autorů uvedeme pouze prvního autora, další spoluautory vynecháme a dopíšeme zkratku „et al.“

- **Název díla** se zapisuje kurzívou.
- **Alternativní** nebo **sekundární odpovědnost** zpravidla se jedná o instituci, která je za vznik publikace zodpovědná, a editory publikace.
- **Označení vydání** je velmi důležité, protože dokument může být časem upraven (rozšířen, zkrácen apod.), a je tedy nezbytné citovat přesnou verzi dokumentu.
- **Místo vydání** uvádíme v podobě, v jaké se objevuje v dokumentu, tj. London, Paris; nepočesťujeme (!) Londýn, Paříž. V případě nejasností, kde se nalézá uvedené místo, je možné do kulatých závorek napsat upřesňující údaj, jako například název státu. Pokud je více míst vydání, zapisujeme první místo v pořadí. Pokud místo vydání není uvedeno, v tomto případě uvedeme zkratku „s.l.“ (sine loco = bez místa). Zkratku umístíme do hranatých závorek.
- **Jméno nakladatele** zapisujeme v co nejkratší srozumitelné podobě. Z názvu vynecháváme zkratky obchodních organizací typu s.r.o., Inc. nebo Ltd..
- **Rok vydání** (tj. datum vytvoření publikace za účelem zpřístupnění). Zpravidla uvádíme rok, u seriálových publikací i měsíc.
- **Rozsah díla** u monografií uvádíme počet stran, u částí sborníků, článků apod. rozsah stran od-do ve svazku.
- **Edici** zapisujeme následujícím způsobem: název edice; číslování v rámci edice.
- **Poznámky:** dodatečné poznámky k publikaci.
- **Standardní číslo:** Čísla ISBN a ISSN zapisujeme na konec záznamu. Tato čísla, pokud se vyskytují v dokumentu, je nutné zapsat, protože jednoznačně identifikují dokument.
- **Lokace ve zdrojovém dokumentu** zpravidla rozmezí stránek od-do nebo název kapitoly. Uvádí se u publikací, které jsou součástí většího svazku.

Příklady knih:

KOSEK, Jiří. *Html – tvorba dokonalých stránek: podrobný průvodce*. Ilustroval Ondřej Tůma. 1. vyd. Praha: Grada, 1998. 291 s. ISBN 80-7169-608-0.

HAUGELAND, John. *Artificial intelligence: the very idea. 3rd printing*, 1987. Cambridge (Massachusetts):MIT Press; London: Bradford Book, c1985. ISBN 0-262-08153-9.

Příklady konferencí:

IWAZUME, Michiaki – TAKEDA, Hideaki – NISHIDA, Toyooki. Ontology-based Information from the Internet. In *Knowledge organization and change: proceedings of the fourth international ISKO conference 15-18 July, 1996 Washington, DC. USA*. Frankfurt am Main: INDEKS, 1996. s. 261–272. (Advances in knowledge organization; vol. 5, ISSN 0938-5495).

Příklady časopisů:

SMEJKAL, V. Proč nový zákon?. *CHIP: magazín informačních technologií*, listopad 1999, roč. 9, c 11, s. 54–55.

GRIFFITH, A. Software publishers association sales data programs: Methodologies, coverage and assumptions. *Publishing Research Quarterly*, 1996, vol. 12, no. 2, s. 21–

*Styl IEEE:*

Struktura citace:

Knihy a části knih:

Autor/editor, “Název: Podnázev,” in *název svazku*, edice, série, vol.. Adresa vydavatele: Vydavatel, Rok, rozsah díla. Available: url dokumentu

Příspěvek ve sborníku:

Autor/editor, “Název: Podnázev,” in *název konference*, místo konání konference,, Rok, stránky. Available: url dokumentu

Články:

Autor/editor, “Název: Podnázev,” název časopisu, vol., no., pp. rozsah díla, Rok a měsíc. Available: url dokumentu

Pozn. Podtržené položky jsou povinné stejně jako psaní názvu kurzívou.

- **Autor** se uvádí ve formě:

*iniciály křestního jména příjmení*

Jména se uvádí bez titulu. Je-li v dokumentu uvedeno více autorů, uvádíme v takovém případě všechny autory do počtu tří. Nad počet tří autorů uvedeme pouze prvního autora, další spoluautory vynecháme a dopíšeme zkratku „et al.“

- **Název** se zapisuje do uvozovek.

- **Název svazku** kurzívou.

- **Rok vydání** (tj. datum vytvoření publikace za účelem zpřístupnění). Zpravidla uvádíme rok, u seriálových publikací i měsíc zkráceně anglicky.

- **Rozsah díla** u monografií uvádíme počet stran, u částí sborníků, článků apod. rozsah stran od-do ve svazku.

U formátu IEEE se neuvádí standardní číslo ISBN a ISSN. Více podrobností o stylu IEEE lze najít v [8].

Příklady knih:

B. Klaus and P. Horn, *Robot Vision*. Cambridge, MA: MIT Press, 1986.

L. Stein, “Random patterns,” in *Computers and You*, J. S. Brake, Ed. New York: Wiley, 1994, pp. 55-70.

R. L. Myer, “Parametric oscillators and nonlinear materials,” in *Nonlinear Optics*, vol. 4, P. G. Harper and B. S. Wherret, Eds. San Francisco, CA: Academic, 1977, pp. 47-160.

Příklady konferencí:

G. R. Faulhaber, “Design of service systems with priority reservation,” in *Conf. Rec.*

1995 *IEEE Int. Conf. Communications*, pp. 3–8.

S. P. Bingulac, “On the compatibility of adaptive controllers,” in *Proc. 4th Annu. Allerton Conf. Circuit and System Theory*, New York, 1994, pp. 8–16.

W. D. Doyle, “Magnetization reversal in films with biaxial anisotropy,” in *1987 Proc. INTERMAG Conf.*, pp. 2.2-1–2.2-6.

Příklady článků:

R. E. Kalman, “New results in linear filtering and prediction theory,” *J. Basic Eng.*, ser. D, vol. 83, pp. 95-108, Mar. 1961.

J. U. Buncombe, “Infrared navigation—Part I: Theory,” *IEEE Trans. Aerosp. Electron. Syst.*, vol. AES-4, pp. 352–377, Sept. 1944.

H. Eriksson and P. E. Danielsson, “Two problems on Boolean memories,” *IEEE Trans. Electron. Devices*, vol. ED-11, pp. 32–33, Jan. 1959.

*Implementace stylů:*

U publikací, u kterých nejsou známy nebo neexistují povinné položky zvýrazněné podtržítkem, tyto položky vynechávám. Například podle rozdělení BibTeXu můžeme mít publikaci typu booklet (brožura) bez autorů nebo prostě autory u některých publikací ani neznáme. Ačkoliv nám normy předepisují povinnost jejich zobrazení v citaci, nelze tento fakt brát jako dogma. Proto jsem to při psaní šablon zohlednil a vlastnosti, které publikaci chybí, jednoduše vynechávám, aniž by to narušilo vzhled citace (např. přebývajícými čárkami, mezi nimiž by mělo být chybějící pole atd.).

O samotné generování citací se stará třída *CitGenerator* ze souboru `/common/CitGenerator.inc.php`. Jde v podstatě o plug-in do systému Smarty, který byl už v původní aplikaci. Tento plug-in umožňuje uložení smarty šablon v databázi. Musel jsem tento modul však upravit, aby byl univerzálnější (předával do šablon všechny proměnné publikace, které byly na vstupu) a předával do šablon autory tak, aby byla šablona schopna zjistit, zda předaná proměnná je příjmení nebo jméno.

Do šablon se předávají proměnné pojmenované podle BibTeXových polí – viz kapitola 10.3. Do šablon se také předávají tzv. rozšiřující atributy, pokud nějaké daná publikace obsahuje.

#### 4.4 Implementace rozhraní pro zadavatele.

Rozhraní pro zadavatele prošlo značnými změnami hlavně kvůli tomu, že jsem implementoval rozdělení na typy publikací, jak je dělí systém BibTeX. Původní rozhraní také trpělo tím, že se v něm problematicky přiřazovali autoři publikací a nebylo možno určit jejich pořadí; dále přibýly nové číselníky „journal“ a „conference“.

Rozhraní pro zadavatele se nachází v adresáři `/load/`, vstupním bodem aplikace je soubor `index.php`, který používá třídu *Load*. Konstruktorem se předávají pole POST a GET ošetřené na výskyt nebezpečných znaků. Struktura třídy *Load* je podobná jako u původní aplikace. Veřejná metoda této třídy se jmenuje `execute`. Jejím voláním se spustí potřebná akce na základě uživatelem zadaných parametrů.

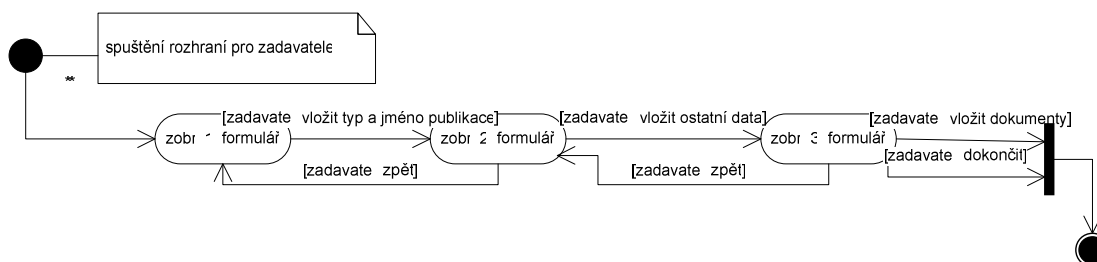
Rozhraní pro zadavatele se opět skládá ze 3 po sobě jdoucích formulářů. V 1. zadavatel

vybere typ publikace, vloží jméno publikace, případně abstract a zařadí publikaci do kategorií.

Vzhled 2. formuláře se odvíjí od typu publikace. Zobrazí se ta pole a číselníky, které lze u publikace vyplňovat. Pole označená hvězdičkou zadavateli signalizují, že jsou povinná. Nicméně ne doslova. Zadavatel je nemusí vyplnit. Pokud je nevyplní (nebo vyplní chybně, např. chybný formát ISBN), bude o tom pouze informován krátkým reportem ve 3. formuláři.

Ve 2. formuláři má zadavatel možnost vybrat položky číselníků *conference* (pro publikace typu InProceedings a Proceedings), *journal* (pro publikace typu Article) a položky číselníku *publisher*. Položky těchto číselníků může přidávat, ale i editovat. Aby byla editace co možná nejpřehlednější, implementoval jsem k tomu účelu AJAXový modul. Tento modul je dobrý k tomu, že když zadavatel vybírá ze seznamu položek číselníku, potřebuje vědět i ostatní doplňující údaje o dané položce. Například pokud vybírá konferenci, zajímá ho např. ve kterém roce a na jakém místě se odehrála. Toto zobrazování podrobností se děje díky Ajaxu bez refreshování stránky. Detailněji se tomuto modulu věnuji v kapitole 4.6.1.

Poslední – 3. formulář zobrazuje celkový souhrn dat, která zadavatel vložil, validační report systému BibTeX a vstupní pole, kam může zadavatel vložit patřičné dokumenty. Dokumentů může vložit libovolně mnoho – stejně jak to šlo u původní databáze publikací. Na 3. formuláři je tlačítko „finish“, po jehož stisknutí se publikace uloží do databáze.



Obrázek 15 - Zadání nové publikace

Jak je vidět, ve třetím formuláři zadavatel vkládá dokumenty PDF (PS a jiné). Text z těchto dokumentů je zajímavý z hlediska pozdějšího fulltextového vyhledávání. Tento text je potřeba nějakým mechanismem z dokumentů vyextrahovat a uložit do tabulky *documents*, na níž bude později aplikováno fulltextové vyhledávání.

Snažil jsem se nalézt nějaký elegantní způsob, jak extrahovat z dokumentů pdf prostý text. Snažil jsem se nalézt knihovnu, která by toto umožňovala. Nakonec jsem zjistil, že jediný způsob jak extrahovat text je, použití externího programu, který je volán ze skriptu aplikace. To poněkud komplikuje pozdější nasazení aplikace, neboť je potřeba tyto externí programy nějakým způsobem nainstalovat; ne každý operační systém bude některé programy podporovat. Je také potřeba nastavit práva na spouštění těchto programů ze skriptu PHP. Tyto požadavky naprosto znemožňují plné nasazení databáze publikací na veřejném hostingu. Aplikaci bude možno sice používat, ale bude fungovat pouze fulltextové vyhledávání v názvu publikací a nikoliv v textu dokumentů.

Program, který se mi jeví být vhodný pro extrahování souborů PDF, se jmenuje XPDF [9]. XPDF je šířen pod licencí GNU GPL v. 2. Na webové stránce projektu je možno získat distribuce pro platformy x86 Linux, x86 Win32 a SPARC Solaris 2.x.

Jediný vhodný program pro extrakci souborů PS je Ghostscript [10]. Předtím jsem se



snažil zkompileovat jiný PS konvertor, určený původně pro operační systém VMS, protože se mi zdál být kompaktnější než Ghostscript. Kompilace se nakonec podařila, ale zjistil jsem, že program nefunguje vždy bez chyby, na některých PS souborech prostě zkolaboval, pro úplnost alespoň uvádím odkaz [11]. Nakonec se tedy ke konverzi PS souborů do textové podoby bude používat program Ghostscript, který je stabilní a pro tento účel velmi vhodný.

#### *Extrakce textu s použitím externích programů:*

Každý z uvedených programů je volán jinak, jsou mu předávány různé parametry, i výstupy těchto programů jsou odlišné, včetně chybových kódů apod.. Proto jsem se rozhodl napsat samostatný modul, který tyto nároky bude řešit, a bude nabízet jednoduché API ostatním částem aplikace.

Tento modul plní funkci jakéhosi adaptéru mezi vlastní aplikací a externím programem. Je uložen v souboru `/common/Convertor.inc.php` a je jím třída *Convertor*. Tato třída obsahuje funkcionalitu: Rozpoznání typu dokumentu, podle něhož vybere příslušný externí program a dále odešle dokument ke zpracování externím programem. Získá textovou podobu dokumentu a předá ji nadřazené logice.

Parametry, které do modulu *Convertor* vstupují, jsou názvy souborů dokumentů a výstupem je jejich textová podoba. Vnitřní logika modulu *Convertor* se stará o správné předání parametrů a získání výstupů. Veřejné metody, které *Convertor* nabízí, se jmenují `add_file` a `finish`. První vezme jako parametr název souboru dokumentu a uloží výsledek konverze do bufferu a druhá uloží obsah bufferu do databáze. Metodu `add_file` je možné volat vícekrát. Po zavolání metody `finish` se pak uloží do tabulky *documents* textový obsah několika dokumentů najednou.

#### *Využití extrakce textu k získání abstraktu:*

Text získaný z dokumentu by se hodil nejen pro potřeby fulltextového vyhledávání, ale šlo by z něj nějakým způsobem získávat abstrakt publikace a ten vkládat do databáze. Moje představa byla původně taková, že na 1. formuláři zadavatelského rozhraní, tam kde je pole na vložení abstraktu, by kromě tohoto pole bylo ještě zaškrtačací políčko, které kdyby uživatel zaškrtl, extrahoval by se abstract automaticky z dokumentu.

Je to dobrá myšlenka a dokonce se používá u nové betaverze známého vyhledávacího katalogu publikací CiteSeer [19]. Když jsem začal experimentovat s touto novou funkcí, zjistil jsem, že programy na konverzi textu neinterpretují správně význam sloupců v dokumentu. Když jsem sledoval chování katalogu CiteSeerX, zjistil jsem, že se chová podobně.

Velmi často se mi stávalo, že abstrakt získaný z dokumentu byl jen směsí necelých vět z jednotlivých sloupců na začátku dokumentu. Program `xpdf` nerozpozná, že text je rozdělen do sloupců, a místo toho čte po řádcích. Ghostscript, který používám na extrakci textu z dokumentů PS má podobný problém, a proto pravděpodobně vznikl program `Prescript` [14], který by měl (?) tento problém řešit. `Prescript` je napsán v jazyce `python` `postscript`. To zase komplikuje jeho nasazení atd..

Kvůli výše zmíněným komplikacím jsem funkci automatického získávání abstraktu nakonec naneštěstí nedodělal. Ponechávám špinavou práci – zadávání abstraktu publikací na samotném uživateli.

### *Paralelní přístup uživatelů:*

V nové implementaci jsem se snažil zajistit, aby současná práce více uživatelů aplikace nezpůsobovala kolize. Problém, který měla původní implementace byl ten, že pokud zadávalo publikace více uživatelů nebo do databáze dokonce zasahoval administrátor, mohlo se někdy stávat, že se zadavatelé např. snažili přiřazovat k publikaci vydavatele, nebo autory, které těsně předtím někdo smazal. Výsledkem bylo strohé chybové hlášení a pád aplikace.

Tomuto problému se prakticky nedá zamezit. Když 1 uživatel přidává publikaci, vybere nějaké položky číselníků z databáze a přiřadí je publikaci ve 2. formuláři, mezi zadáváním publikace a jejím uložením uběhne dlouhá doba, za kterou někdo jiný může položky číselníků prostě smazat a při finálním vkládání publikace dojde k chybě.

Proto jsem v nové implementaci za začátek transakce vložil kontrolní rutinu, která zjistí, zda požadovaní autoři, vydavatelé, časopisy nebo konference existují. Pokud ano, transakce doběhne do konce. Pokud ne, zobrazí se zadavateli přehledné chybové hlášení, které ho informuje o tom, co konkrétně se smazalo a vybídne ho, aby situaci vyřešil.

Jiná situace by to byla, pokud by někdo nikoli smazal, ale změnil položku číselníku. Tohoto problému se však není třeba moc obávat. Jediný, kdo může libovolně měnit položky číselníků je administrátor, který by si měl být vědom rizika, které by mohl způsobit. Ostatní uživatelé mají editaci číselníků většinou povolenou jen z části. Nemohou měnit hlavní vlastnosti položek číselníků jako je třeba název konference a rok, název časopisu atd..

Standardní úroveň izolace pro MySQL InnoDB je repeatable read. Není potřeba nastavovat úroveň vyšší. V této aplikaci se nepotkáme s případy, kdy by bylo potřeba řešit phantom effect.

Ve víceuživatelských systémech se můžeme potkat s těmito anomáliemi:

- Lost update (závislost write / write)
- Dirty read (závislost write / read)
- Inconsistent analyses (závislost read / write)
- Phantom effect

Phantom effect je vlastně jakási write / read závislost na vyšší úrovni

**Phantom effect** může nastat např. v tomto scénáři:

1. Transakce T1 přečte množinu záznamů, které vyhovují nějaké konkrétní podmínce.
2. Transakce T2 potom vloží jeden nebo více záznamů, které vyhovují této podmínce a volá COMMIT.
3. T1 čte množinu záznamů s tou samou vyhledávací podmínkou a obdrží jak původní záznamy, tak záznamy vložené transakcí T2.

## 4.5 Implementace importu ze systému BibTeX

Import publikace z BibTeXové definice je pohodlný způsob jak nahrát do databáze publikaci převzatou z webových katalogů. Skripty rozhraní pro import jsou v adresáři /import/. Veškerá logika importního formuláře je ve třídě *Import* (soubor *Import.inc.php*) a vzhled formuláře v šabloně *import.tpl*. V Souboru *Lexan.inc.php* je lexikální analyzátor. V souboru *Parser.inc.php* se nachází parser.

Než se budu věnovat samotné implementaci, popíšu formát BibTeX. Zde je ukázkový příklad jedné BibTeXové definice:

```
@inproceedings{1269060,  
  author = {Jing Dong and Yajing Zhao},  
  title = {{E}xperiments on {D}esign {P}attern {D}iscovery - {EDFP}},  
  booktitle = {PROMISE '07: Proceedings of the Third International  
Workshop on Predictor Models in Software Engineering},  
  year = {2007},  
  isbn = {0-7695-2954-2},  
  pages = {12--25},  
  publisher = {IEEE Computer Society},  
  address = {Washington, DC, USA}  
}
```

Definice hned na začátku určuje typ publikace. Závorka ‘{’ označuje začátek definice atributů. Hned za ní následuje tzv. citační klíč. Jména atributů vlevo jsou klíčová slova systému BibTeX. Klíčovým slovům je za znakem ‘=’ přiřazena hodnota ve složených závorkách nebo uvozovkách. Definice atributů jsou od sebe odděleny čárkou.

Za povšimnutí stojí to, že hodnoty atributů mohou v sobě obsahovat další složené závorky, což se používá k zachování velkých písmen. BibTeX totiž někdy převádí textové hodnoty na malá písmena. Rozsah stránek se zapisuje použitím dvojité pomlčky.

Jména autorů se zapisují dvěma způsoby. Nejčastěji se píše první jméno a příjmení na 2. místě – stejně jak je tomu v uvedeném příkladě. Druhý možný způsob zápisu: *příjmení, jméno* se používá méně. Prostřední jméno, je-li definováno, se pak zapisuje také za čárku hned po křestním jménu. Jednotlivé autory od sebe odděluje slovo „and“.

Navrhl jsem jednoduchý parser LL(1) gramatiky dle [12], který pomocí rekurzivního sestupu získá hodnoty jednotlivých BibTeX atributů a dále je předá ke zpracování proceduře, která vloží novou publikaci do databáze.

*Lexikální elementy:*

```
pub_type = '@article' | '@book' | '@booklet' | '@inbook' | '@incollection' |  
'@inproceedings' | '@manual' | '@mastersthesis' | '@misc' | '@phdthesis' |  
'@proceedings' | '@techreport' | '@unpublished'
```

```
keyword = 'address' | 'annotate' | 'author' | 'booktitle' | 'chapter' | 'crossref' | 'edition' |  
'editor' | 'howpublished' | 'institution' | 'journal' | 'key' | 'month' | 'note' | 'number' |  
'organization' | 'pages' | 'publisher' | 'school' | 'series' | 'title' | 'type' | 'volume' | 'year' |  
'affiliation' | 'abstract' | 'contents' | 'copyright' | 'isbn' | 'issn' | 'keywords' | 'language' |  
'location' | 'lccn' | 'mrnumber' | 'url' | 'doi'
```

number = číslice {čísllice}

strval = lpar abb rpar | “ abb “

abb = písmeno | číslice {písmeno | číslice}

str = 'a' | 'b' | ... | 'z' | 'A' | 'B' | ... | 'Z'  
 num = '0' | '1' | '2' | '3' | '4' | '5' | '6' | '7' | '8' | '9'  
 lpar = '{'  
 rpar = '}'  
 comma = ','  
 assign = '='

BibTeXovou definici parsuji podle následující atributované gramatiky, uvádím zde množiny FIRST a FOLLOW pro neterminály a rozkladovou tabulku, podle které jsem naprogramoval rekurzivní sestup ve třídě *Parser*.

Atributovaná gramatika:

1.	Bibtex -> pub_type lpar Def rpar	
2.	Def -> Equation RestDef	
3.	Def -> abb comma Equation RestDef	
4.	Def -> number Equation RestDef	
5.	Equation -> keyword assign RestEq	RestEq.dname := keyword.sname
6.	RestDef -> comma Equation RestDef	
7.	RestDef -> $\epsilon$	
8.	RestEq -> number <b>declBib</b>	declBib.dname = RestEq.dname declBib.dval = number.sval
9.	RestEq -> strval <b>declBib</b>	declBib.dname = RestEq.dname declBib.dval = strval.sval

Tabulku atributů gramatiky:

	název symbolu	dědičné atributy	syntetizované atributy
Neterminály	Equation		sname
	RestEq	dname	
vstupní symboly	strval		sval
	number		sval
	abb		sval
speciální výstupní symboly	declBib	dname, dval	

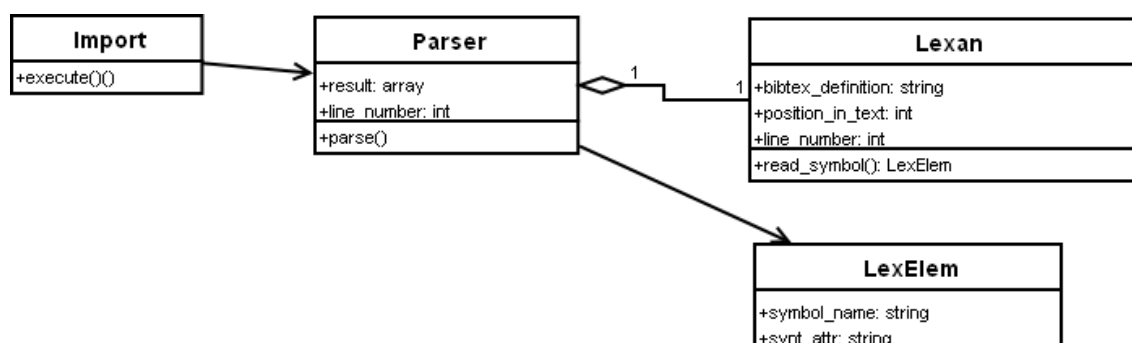
Množiny FIRST a FOLLOW:

	FIRST	FOLLOW
Bibtex	pub_type	$\epsilon$
Def	keyword, abb, number	$\epsilon$ , rpar
Equation	keyword	$\epsilon$ , rpar, comma
RestDef	comma, $\epsilon$	$\epsilon$ , rpar
RestEq	number, strval	$\epsilon$ , comma, rpar

Rozkladová tabulka  $LL(1)$  gramatiky:

	pub_type	abb	strval	number	keyword	lpar	rpar	comma	e
Bibtex	1								
Def		3		4	2				
Equation					5				
RestDef							7	6	7
RestEq			9	8					

Speciální neterminální symbol *declBib* slouží k deklaraci BibTeXového pole, jsou mu předány dědičné atributy název BibTeXového pole a hodnota pole. Hlavním programovým modulem je třída *Parser* a vstupním bodem její metoda *parse*. *Parser* si vytvoří lexikální analyzátor *Lexan*. Při parsování textu pak parser volá metodu *read\_symbol* třídy *Lexan* a získává lexikální elementy jako objekty třídy *LexElem*. Třída *Lexan* při čtení vstupního textu počítá pozici a číslo řádku, na kterém se právě nachází a předává je i syntaktickému analyzátoru. Vyskytne-li se ve vstupním textu chyba, zobrazí se hlášení s popisem a číslem řádku, kde se chyba nachází. Diagram tříd viz následující obrázek.



Obrázek 16 - Parser

Lexikální analyzátor se stará také o to, aby hodnota elementu *strval* neobsahovala složené závorky, které se používají k zachování velikosti písmen. Pokud řetězec tyto závorky obsahuje, lexikální analyzátor je přeskočí. Je schopen interpretovat definice polí ve tvaru

pole = "text",

ve tvaru

pole = {text},

dokonce i ve tvaru

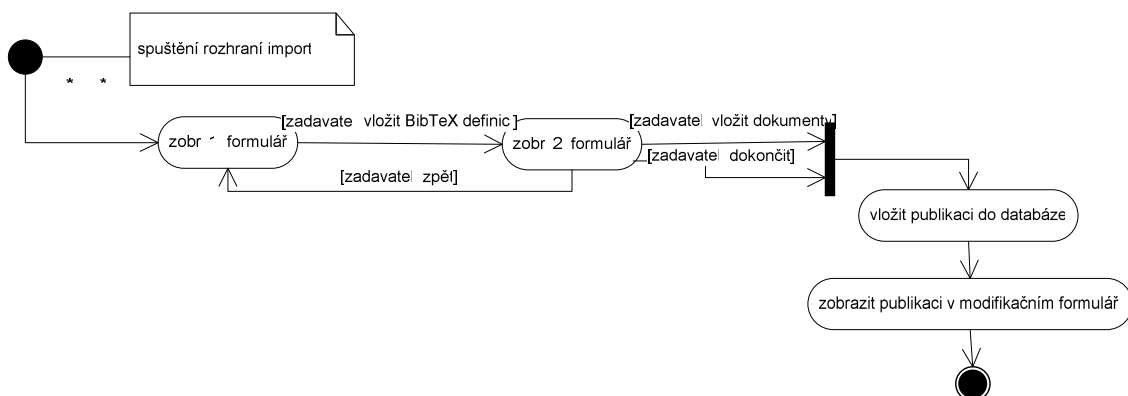
pole = číslo.

Poslední tvar definice se používá někdy u číselných údajů např. lze psát *year* = 2008 namísto *year* = "2008".

Parser má na starosti kromě samotného parsování definice také rozklad pole autorů z podoby „forenename surname and forename2 surname2 and ...“ nebo „surname, forename and surname2, forename2 and ...“ do struktury, ze které může potom třída *Import* získat jména a příjmení autorů.

### Import publikace:

Proces importu probíhá následovně: Zadavatel do textového vstupu vloží definici publikace ve formátu BibTeX a může vybrat kategorie. Odešle formulář a zobrazí se mu seznam nalezených BibTeXových polí a validační report. Validační report pouze upozorní zadavatele, pokud nějaká povinná pole chybějí. Stejně, jak tomu je i v rozhraní pro ruční zadání publikace. Zadavatel může a nemusí vložit dokumenty pdf, ps atd.. V dalším kroku aplikace informuje uživatele o úspěšném importu a zobrazí importovanou publikaci ve formuláři, kde může uživatel provést ještě dodatečné úpravy.



Obrázek 17 - Import publikace

Vložení publikace do databáze má na starosti metoda finish třídy *Import*. V této metodě se rozhoduje, jak budou k publikaci přiřazeny číselníky autor, vydavatel, časopis a konference. Neexistuje-li požadovaná položka číselníku v databázi, jednoduše se vytvoří, existuje-li pouze se k publikaci přiřadí. To, jestli daná položka existuje, se u různých číselníků zjišťuje různě. Celý proces vkládání publikace je uzavřen v transakci.

Přiřazení číselníků publikacím je možné změnit na konci importu. Aplikace automaticky přesměruje zadavatele na formulář kde může dodatečně editovat vlastnosti importované publikace.

Do databáze se ukládají všechny „standardní“ atributy systému BibTeX kromě *annotate*, *key* a *crossref*, které pro nás nemají význam. Z nestandardních jsou podporovány atributy *location*, *isbn*, *issn* a *url*.

A autorů se kontroluje, zda souhlasí jméno, prostřední jméno a příjmení. U vydavatelů se kontroluje název a adresa. U konferencí se kontroluje název a rok, u časopisů název a ISSN.

Rozsah stránek z BibTeXové definice v podobě `pages = "123--321"` se převede na řetězec s jednou pomlčkou.

Název měsíce může být v BibTeXové definici zadán názvem a nebo číslem. Parser rozpozná oba způsoby, dokonce akceptuje i zkratky názvu měsíce jako „jan – january“.

### Paralelní přístup uživatelů:

Vložení publikace, včetně vytváření položek číselníků a jejich přiřazování publikaci, je uzavřeno v transakci. Tím se import liší od klasického ručního zadávání publikace, kde zadavatel nejprve vybíral položky číselníků. Na konci procesu na posledním formuláři,

když chtěl dokončit vkládání publikace, mu mohla aplikace zhavarovat se zprávou, že položku číselníku mu někdo smazal.

U importu je tomu jinak. Rozhodování o tom, které položky číselníků se publikaci přiřadí nebo které se vytvoří a přiřadí, se děje v bloku transakce. MySQL používá jako výchozí úroveň izolace repeatable read, která postačuje, neboť v našem případě není potřeba řešit tzv. phantom effect.

## 4.6 Implementace administračního rozhraní

Administrační rozhraní bylo potřeba upravit tak, aby reflektovalo zavedení nových typů publikací převzatých ze systému BibTeX, aby formulář pro editaci publikací byl intuitivnější a bylo možno spravovat nové číselníky (časopisy, konference). Dále bylo potřeba vytvořit formulář pro správu uživatelů aplikace. Zrušil jsem ovládací prvky, kterými administrátor schvaloval publikace, autory, vydavatele atd.. Ponechal jsem však seznam publikací rozdělen na dvě části; na publikace nové a zkontrolované. Zadavatelem vložené nebo upravené publikace se zobrazují administrátorovi jako nové.

Programová struktura administračního rozhraní zůstala v podstatě stejná, pouze přibýly v administračním rozhraní nové záložky pro časopisy, konference a uživatele. Administrační rozhraní nyní používá celkem 10 záložek. Každá je reprezentována třídou, která dědí od abstraktní třídy *Admin*. Skripty administračního rozhraní se nacházejí v adresáři /admin/. Konkrétní záložku lze vytvořit pomocí statické factory metody *create* ze třídy *Admin*, parametrem této metody je číslo záložky. Šablony, které administrační rozhraní používá se jmenují *admin\_tabXX.tpl* a *modify\_publication.tpl*.

### 4.6.1 Formulář pro modifikaci publikace

Tento formulář lze vyvolat z první nebo druhé záložky administračního rozhraní přes tlačítko „modify“. Jeho vzhled je definován šablonou *modify\_publication.tpl* a klientský skript je uložen v souboru /admin/modify\_publication.js. Tímto formulářem lze editovat veškeré vlastnosti publikace, od jejího typu až po kategorie do kterých patří.

Vylepšil jsem způsob jakým může administrátor pro publikaci vybírat autory a definovat zároveň jejich pořadí. Tuto funkci jsem implementoval i v rozhraní pro zadavatele.

Původní problém s vybíráním autorů jsem vyřešil tak, že zobrazuji 2 pole *select*. V prvním je úplný seznam všech autorů, kteří jsou v databázi a lze je přidávat nebo odebírat z/do druhého seznamu. Edituje-li administrátor autory publikace a rozmyslí-li si, že editovanou publikaci neuloží, žádné změny se do databáze nepromítnou, protože seznam vybraných autorů existuje pouze dočasně v *session* a po odchodu na jiný formulář se smaže.

Změnou typu publikace se samozřejmě změní i nároky na její atributy. Proto jsem napsal klientský skript, který okamžitě skrývá pole, která jsou pro určitý typ publikace nesmyslná, a naopak zobrazuje pole, která mají význam.

*Položky číselníků:*

Značná potíž byla však v tom, jak umožnit editaci číselníků časopis, konference a

vydavatel. Administrátor má mít možnost vybrat nějakou položku ze seznamu. Tato položka má nějaký název, ale za ním se skrývají další údaje (např. adresa u vydavatele, místo konání konference, ISSN časopisu atd.). Možnosti jak umožnit jejich editaci jsou dvě. První je vytvořit pro položky číselníků vlastní formuláře a na ně odkazovat administrátora, když vybere editaci dané položky. Druhá je s použitím technologie Ajax.

Ajax je pro tento případ vhodnou volbou. Napsal jsem javascriptové funkce, které zasílají asynchronní požadavky na server při každé události `onchange` html prvků `select` zobrazujících položky číselníků.

Při každé takové události se vytváří objekt `XMLHttpRequest`, který vysílá požadavek na server a při příchodu odpovědi se vyvolá event handler, který je svázán s vlastností `onreadystatechange` objektu `XMLHttpRequest`. V tomto handleru je kód, který zobrazí získané údaje v prohlížeči.

Všechny tyto asynchronní požadavky obsluhuje skript `/admin/detail_provider.php`, který si pro svoji potřebu vytváří instanci třídy `AjaxDetailHandler` (ze souboru `/common/AjaxDetailHandler.inc.php`), které předává parametry již ošetřené na výskyt znaků jednoduchá uvozovka (`'`), dvojitá uvozovka (`"`) a zpětné lomítko (`\`).

Parametry, které lze skriptu `detail_provider.php` předávat metodou GET:

`publisher.....id vydavatele`

`conference.....id konference`

`journal.....id časopisu`

Odpověď, kterou `AjaxDetailHandler` vygeneruje, nejsou v tomto případě přímo data XML, ale pouze definice javascriptového pole, která se potom interpretuje na straně klienta.

V případě požadavku na vydavatele vrátí `AjaxDetailHandler` 3 prvkové pole s hodnotami `{used, name, address}`. Dotaz na časopis získá jako odpověď pole `{used, name, ISSN}`. Prvek „used“, zde značí, že položka je už používána (přiřazena) některým publikacím.

Dotaz na konferenci dostane jako odpověď JS pole s položkami `{allowed, name, abbreviation, year, from, to, location, isbn}`. Položka „allowed“ značí, zda je u této konference povolena editace názvu a roku. Konference, u kterých je zakázána editace, jsou buď konference již přiřazené některým publikacím, nebo neúplně zadané konference (chybí rok)<sup>1</sup>.

Klientský skript vyhodnotí parametry, které získal asynchronním voláním a v závislosti na parametrech „used“ (pro vydavatele a časopisy) a „allowed“ (pro konference) zpřístupní/zakáže editaci příslušejících textových polí.

Tato omezení v editaci některých polí se týkají pouze modifikačního formuláře. Administrátor má stále možnost upravovat položky všech číselníků použitím 3. – 8. záložky administračního rozhraní. Omezení platí pouze pro formulář, kterým se modifikuje publikace. Tento formulář mohou totiž využívat i sami zadavatelé publikací. Každý uživatel aplikace s oprávněními role „zadavatel“ má možnost vidět stejný

---

<sup>1</sup> Neúplně zadané konference ze nezobrazují k výběru v rozhraní pro zadavatele a v administračním rozhraní na modifikačním formuláři je však potřeba je zobrazit protože některé publikace na ně můžou odkazovat. Pokud by pak administrátor chtěl neúplnou konferenci k publikaci přiřadit a zároveň nebyl jejím vlastníkem, zobrazí se mu chybové hlášení ihned po vybrání konference ze seznamu.



formulář jako administrátor na 2. záložce detailu publikace.

Položky číselníků konference, časopis a vydavatel je nyní možné intuitivně přidávat, prohlížet i editovat. Ke každému číselníku jsem ještě přidal tlačítko „unset“, kterým se zruší výběr položky. Přenesl jsem tuto AJAXovou funkcionalitu i do rozhraní pro zadavatele. Už při zadávání nové publikace je možné tento komfort použít.

Je zřejmé, že formulář pro modifikaci publikace je vybaven logikou, která se interpretuje na straně klienta. Vypne-li uživatel ve svém prohlížeči podporu javascriptu, přestane tak být formulář „interaktivní“. Nebude moci během modifikování publikace editovat položky číselníků, formulář se nebude přizpůsobovat podle typu publikace, místo toho bude zobrazovat úplně všechny položky. Je potřeba se smířit s tím, že „interaktivita“ může být jen tam, kde jí to uživatelé dovolí.

Nepodporuje-li klientský prohlížeč javascript, formulář přijde o svoji interaktivitu. I přes to ovšem na straně serveru kontroluji, jaká data od uživatele přicházejí a např. u položek číselníků, které nemají být editovány, tuto editaci zakazuji na straně serveru.

#### **4.6.2 Nové záložky**

Vytvořil jsem záložky pro správu časopisů, konferencí a uživatelů. Stejně jak tomu bylo u staré databáze publikací, dávám tím možnost administrátorovi libovolně přidávat, editovat a mazat jakékoli položky bez ohledu na to, zda se tím nějak naruší konzistence publikací či nikoli. U každé položky zobrazuji pouze výstražnou ikonku, která varuje před mazáním záznamů, které jsou používány. V záhlaví datových sloupců se zobrazují odkazy, kterými lze řadit záznamy.

Na záložce uživatelů lze mazat/vytvářet uživatele aplikace. Existujícím uživatelům může administrátor nastavovat úrovně oprávnění administrátor, zadavatel a čtenář.

Více informací o struktuře záložek administračního rozhraní naleznete v dokumentaci původní databáze [1] v kapitole 3.4.

## 5 Zabezpečení

### 5.1 Autentizace uživatelů

Původní aplikace prováděla http autentizaci pomocí metody Basic. Nevýhodou bylo to, že heslo se při každém http požadavku na zaheslovanou stránku opakovaně přenášelo v otevřeném tvaru po síti.

Nově jsem naimplementoval formulářovou autentizaci, kde se heslo přenáší pouze jednou při přihlašování. V instalační příručce popisuji postup jak na webovém serveru zapnout protokol https právě za účelem ochrany uživatelských hesel.

Přihlašování se provádí na formulářích, umístěných v adresáři /account/, který je přístupný jen přes protokol https. Více o novém formulářovém přihlašování popisuji v kapitole 4.2.

.

### 5.2 Ošetření uživatelských vstupů

Stejně jako u původní aplikace, bylo mojí snahou, ošetřit vstup od uživatelů tak, aby se v první řadě zamezilo vzniku rizika skriptové a SQL injekce. V druhé řadě bylo nutné zajistit, aby systém neinterpretovat texty obsahující HTML. V podstatě nejde o nic nového, v některých odstavcích cituji text z [1], pouze cítím povinnost zde zmínit jak je aplikace zabezpečena proti výše zmíněným rizikům.

#### *Script injecion:*

Injekci skriptu jsem vyloučil už od prvopočátku, neboť nikde nedávám útočnickovi možnost, aby přes parametry URL adresy ani POST daty mohl manipulovat s nějakými soubory.

Problém s injekcí skriptu mohl v původní implementaci nastat v případě, že by útočník skript poslal systému prostřednictvím vstupního formuláře k zadávání publikací. Je zde přeci možnost ke každé publikaci přiložit nějaké soubory. V původní implementaci, jsem toto riziko řešil tím, že jsem povolil vkládání souborů s určitými příponami.

V nynější implementaci ještě k tomu zakazuji přístup do adresáře /storage/, který obsahuje uploadované dokumenty souborem „.htaccess“. Souborem .htaccess jsem zakázal přístup i do adresáře /lib/xpdf, který obsahuje konvertor pdf2txt.

#### *SQL injection:*

Dovolím si citovat pár faktů o bezpečnosti PHP v kombinaci s MySQL:

- PHP nepovoluje vykonávání několika SQL dotazů v jednom volání funkce `mysql_query()`, takže tím odpadá problém s tím, že by útočník vložil za prováděný SQL dotaz svůj vlastní kód. Provádí se vše, co se nachází před prvním středníkem.
- V MySQL neexistuje nic, co by mohlo zavolat externí aplikaci jako je tomu například u MSSQL, takže odpadá spousta dalších potenciálně nebezpečných dotazů.
- Pokud v databázi neuchovávám PHP kód který následně pomocí funkce `eval()` vykonávám, nehrozí ani provedení cizího kódu na serveru.

Aby se tedy zamezilo riziku SQL injekce, bylo zapotřebí zajistit proměnné získávané z XHTML formulářů musejí být ošetřeny tak, aby nebylo možné v nich poslat samotné znaky typu uvozovka, dvojité uvozovka.

Tato filtrace se standartě děje na serverech s PHP interpreterem, se zapnutou funkcí magic quotes přidáváním znaku backslash (\) před speciální znaky. Protože se na to ale nedá spoléhat, byl můj kód vybaven vlastní filtrací. Ta funguje tak, že napřed otestuje, jestli jsou magic quotes zapnuté, a když ne, vykoná se na všechny uživatelské proměnné funkce addslashes(). Jelikož se ale přes vstupní formuláře posílají i data typu pole, bylo zapotřebí zajistit, aby se addslashes provedlo i pro všechny prvky pole. K tomu slouží rekurzivní funkce clean\_rec definovaná v souboru /common/misc\_fns.inc.php.

Co se týče dat, která do aplikace vstupují, ty jsou předávány metodou POST a GET. Na tato pole hodnot je vždy zavolána funkce clean\_rec a jejich výsledky jsou předány konstruktorem příslušných tříd které uživatelská data zpracují. Uvnitř objektů se pracuje vždy a pouze s těmito hodnotami a už se nikdy neodkazuje na superglobální proměnné \$\_POST a \$\_GET. Takto získané hodnoty se ve všech případech ještě vkládají do řetězce s SQL dotazem mezi dvojici jednoduchých uvozek.

Při implementaci importu BibTeX definic jsem narazil na problém, že uživatelem zadaná definice obsahuje nechtěné znaky zpětné lomítko (\) před každým znakem uvozovka. Zpětná lomítka přidává samo PHP, když je zapnuta funkce magic quotes. Pokud je tedy tato funkce zapnuta, detekuji to a zpětná lomítka odebíráám funkci stripslashes, abych mohl definici parsovat. Teprve po zpracování definice parserem volám na data vstupující do databáze funkci addslashes.

#### *HTML ve vstupních datech:*

Opomenutí rizika interpretace HTML kódu, který mohl útočník do textu vložit, není tak strašlivé jako hrozba SQL injekce, ale i tak to může být někdy poměrně závažné bezpečnostní riziko.

Filtrovat HTML entity je zapotřebí nejen kvůli útočníkům, ale i kvůli nic netušícím uživatelům, kteří by rádi vkládali znaky třeba „<>” a další jiné. Já jsem problém řešil tak, že data, která uživatel vloží se uchovávají v databázi v nezměněné podobě a teprve při jejich výstupu, je ošetřen výskyt HTML entit. Někde data filtruji PHP funkcí htmlspecialchars, jinde využívám možností systému Smarty a filtraci HTML entit provádím v šablonách.

### 5.3 Chráněné adresáře

#### *Ochrana dokumentů PDF, PS (...) před vykradením:*

Adresář /storage/ obsahuje dokumenty publikací. Vzhledem k tomu, že databáze publikací není veřejná a dokumenty v ní uložené jsou chráněny autorským zákonem, zajistil jsem, aby je neautorizovaný uživatel nemohl stáhnout. Přístup k souborům pomocí http GET jsem zakázal v souboru .htaccess a umožnil stažení těchto souborů pouze pomocí skriptu, který ověřuje identitu uživatele. Více podrobností o tomto mechanismu jsem popsal na konci kapitoly 3.3.3.

## 6 Testy

Aplikaci jsem testoval ručním vložením 20 publikací všech typů a 20 importy BibTeXových definic převzatých hlavně z ACM portálu. Je třeba podotknout, že ACM portál exportuje publikace s chybou (na konci nefinice vždy přebývá čárka).

Provedl jsem testy zabezpečení vkládáním speciálních znaků (‘, “, \) a HTML entit.

Na všech stránkách jsem provedl test validity XHTML kódu. Jako validátor jsem použil Tidy Validator 0.8.5.2, který jsem si nainstaloval jako plugin [29] do prohlížeče Mozilla Firefox. Testy jsem provedl s nastavenou úrovní přístupu „normal“. XHTML kód byl označen bezchybným, validátor většinou nehlásil žádné chyby ani varování. Výjimkou byla hlavní strana, kde používám nestandardní atribut „autocomplete“ u elementu input pro potřeby našeptávače.

Aplikaci jsem testoval na prohlížečích Mozilla Firefox 3.0.5, Opera 9.6 pro Windows a Microsoft Internet Explorer 6, v rozlišeních od 1024x768 do 1600x1200. V těchto prohlížečích funguje bez problémů. Následně jsem zkusil použít prohlížeč Mozilla Firefox, u kterého jsem vypnul interpret javascriptu. Dokázal jsem, že bezpečnost aplikace není absencí javascriptu ohrožena.

## 7 Porovnání s existujícími systémy

Nejbližší má moje databáze publikací k systému Aigaion, který popisují v kapitole 3.1.1. S komerčními systémy ji ani srovnávat nemohu, protože jsou o poznání propracovanější a zpravidla na jejich vývoji pracuje tým odborníků řadu let. Za účelné považuji srovnání s webovými aplikacemi Aigaion a BibAdmin.

### *Porovnání se systémem Aigaion:*

Mojí databázi publikací jsem vyvinul za stejným účelem, jako byl vyvinut systém Aigaion. V podstatě šlo o to, vytvořit systém ve kterém budou publikace uloženy. Aby bylo možno je třídit do kategorií a podle nich vyhledávat. Aby šlo hledat i podle dalších kritérií jako např. dle názvu, jmen autorů atd. Důraz je kladen na to, aby se usnadnila práce vědcům, kteří píšou odborné publikace a ke své práci využívají zdroje, které ve svých dílech citují.

Systém Aigaion se v porovnání s mojí databází publikací snadněji instaluje. Nevyžaduje totiž, aby byl na serveru instalován PEAR framework. Nepoužívá žádné externí programy pro konverzi textu z dokumentů PDF/PS. Jediné co ke své funkci potřebuje je databáze MySQL. Na druhou stranu zase Aigaion nenabízí vyhledávání ve fulltextu dokumentů.

Aigaion jemněji dělí uživatelská práva, má 2<sup>12</sup> možných druhů oprávnění. Definuje také úroveň přístupnosti pro jednotlivé publikace; tyto úrovně se vztahují se na práva čtení a modifikaci publikace. Naproti tomu moje databáze publikací rozlišuje pouze 3 typy uživatelských oprávnění a neaplikuje žádné restriktce na samotné publikace.

Vývojáři Aigaionu zvolili také výrazně odlišný datový model. Zatímco já ukládám autory, vydavatele, časopisy a konference v samostatných tabulkách, Aigaion ukládá samostatně pouze autory. Oba přístupy mají něco do sebe. Já jsem se přiklonil se svolením vedoucího práce k první variantě.

Aigaion umožňuje navíc oproti mé aplikaci export/import do RIS formátu (EndNote).

Chybí mu ale možnost vyhledávání ve fulltextu publikací.

#### *Porovnání se systémem BibAdmin:*

BibAdmin [19] je systém, který jsem představil už v mé předchozí práci a nyní jej zde srovnávám znovu pro úplnost. Moje aplikace je proti systému BibAdmin mnohem vyspělejší co se týče možností vyhledávání, exportu do standardních citačních formátů a BibTeXových importů. Moje aplikace má intuitivnější administrační rozhraní.

Vyhledávání v BibAdminu je koncipováno trochu odlišným způsobem. Na rozdíl od mé implementace BibAdmin nemůže hledat zároveň v názvech publikací a definovat filtry do jakých kategorií, resp. od jakých autorů se má vyhledávat; a neumí vyhledávat ve fulltextu publikací. BibAdmin má pouze jedno vstupní pole na zadání hledaného textu a selectem vedle něj, se má určit v čem vyhledávat (jména autorů, publikací, konferencí, časopisů a kategorií). Součástí BibAdmina je katalog, ve kterém můžeme dát kliknutím příkaz, aby se například zobrazily publikace z daného roku a jiné.

Nedostatek, kterým BibAdmin trpí je to, že není možné změnit formát citací jinak, než přímou editací zdrojového PHP kódu. Aby BibAdmin plně fungoval, musí být u klienta zapnut javascript. Za větší nedostatek považuji absenci jakékoliv dokumentace, ať už dokumentace kódu nebo instalační či uživatelské příručky. Žádný z těchto textů není k dispozici. PHP kód obsahuje minimum komentářů, nejsou používány objekty ani šablony, HTML a PHP kód jsou kombinovány do sebe, což působí nepřehledně.

Moje implementace umožňuje měnit formát zobrazovaných citací dle šablony, kterou si vybere uživatel sám z nabídky. Administrátor může šablony upravovat, a to bez zásahu do zdrojových kódů aplikace.

## **8 Závěr**

Upravil jsem aplikaci, aby byla schopna rozlišovat 13 typů publikací, podobně jako je rozlišuje systém BibTeX. Tento přístup umožňuje jemněji členit publikace než tomu bylo doposud. Typy atributů, které lze přiřadit publikacím jsem převzal také ze systému BibTeX.

Reflektoval jsem nové požadavky na zlepšení uživatelského rozhraní, které vznikaly během používání původní aplikace. Zpřehlednil a zjednodušil jsem proces zadávání nových publikací. Zjednodušil jsem proces schvalování publikací, který byl v původní implementaci příliš restriktivní.

Rozšířil jsem vyhledávací schopnosti aplikace, která nyní dokáže fulltextově vyhledávat v názvech publikací a v textu přiložených dokumentů PDF i PS. Dále jsem upravil vyhledávání dle autorů, tak aby je bylo možno specifikovat v textové podobě. Vyhledávání podle autorů jsem vybavil Ajaxovým našeptávačem. Výsledky hledání lze filtrovat podle klíčových slov (kategorií) nebo je možné dle samotných kategorií i vyhledávat. Dokonce lze specifikovat, zda nalezené publikace mají být zařazeny do všech nebo jen do některých kategorií, které uživatel označil. Nalezené publikace lze nyní řadit podle několika kritérií.

Implementoval jsem export a import BibTeX definic a standardních citačních formátů IEEE a ČSN.

Implementoval jsem centralizovanou správu uživatelů a lépe jsem zabezpečil mechanismus přihlašování do aplikace. Zabezpečil jsem také úložiště dokumentů proti jejich zcizení.

V průběhu analýzy jsem však narazil na potíže s protokolem Z39.50 (kapitola 2.4), který jsem nakonec neimplementoval.

Zhodnotil jsem možnosti konkurenčních systémů a porovnal jsem svoji implementaci se dvěma systémy, které mají k mé implementaci co do složitosti a zaměření nejbližší.

V poslední řadě jsem převedl publikace, které byly uloženy ve starém datovém modelu do nové struktury databáze.

## 9 Seznam literatury

[1] VODIČKA, Martin. Databáze publikací, bakalářská práce, ČVUT v Praze. 2006.

URL: <[https://dip.felk.cvut.cz/browse/pdfcache/vodicm1\\_2006bach.pdf](https://dip.felk.cvut.cz/browse/pdfcache/vodicm1_2006bach.pdf)>

[2] BibTeX and bibliography styles description on Colorado Univerzity homepage

URL: <<http://amath.colorado.edu/documentation/LaTeX/reference/faq/bibstyles.html>>

[3] Apache SSL project page <http://www.apache-ssl.org>

[4] MySQL fulltext engine manual <http://dev.mysql.com/doc/refman/5.0/en/fulltext-search.html>

[5] AjaxAC framework project page <http://ajax.zervaas.com.au/>

[6] BOLDIŠ, Petr. Bibliografické citace dokumentu podle CSN ISO 690 a CSN ISO 690-2: Část 1 – Citace: metodika a obecná pravidla. Verze 3.3.

URL: <<http://www.boldis.cz/citace/citace1.pdf>>.

[7] BOLDIŠ, Petr. Bibliografické citace dokumentu podle CSN ISO 690 a CSN ISO 690-2: Část 2 – Modely a příklady citací u jednotlivých typů dokumentů. 2004.

URL: <<http://www.boldis.cz/citace/citace2.pdf>>.

[8] IEEE Editorial Style Manual

[http://www.ieee.org/portal/cms\\_docs\\_iportals/iportals/publications/authors/transjnl/styl\\_emanual.pdf](http://www.ieee.org/portal/cms_docs_iportals/iportals/publications/authors/transjnl/styl_emanual.pdf)

[9] XPDF project page <http://www.foolabs.com/xpdf/>

[10] GhostScript project page <http://pages.cs.wisc.edu/~ghost/>

[11] Ps2txt for VMS utility <http://mvb.saic.com/freeware/freewarev40/ps-to-text/>

[12] Müller, Karel. *Programovací jazyky*; [s.l.]: Nakladatelství ČVUT, 219 s., Skriptum, ISBN 80-01-02458-X.

[13] GARRETT, Jesse James, Ajax: A New Approach to Web Applications

URL: <<http://www.adaptivepath.com/ideas/essays/archives/000385.php>>

[14] Prescript project page <http://www.nzdl.org/html/prescript.html>

[15] RUBRINGER, Tomáš. Z39.50. *Ikaros* [online]. 1999, roč. 3, č. 8 [cit. 2009-01-22].

URL: <<http://www.ikaros.cz/node/1034>>. ISSN 1212-5075.

[16] Z39.50 Profiles specification

<http://www.loc.gov/z3950/agency/profiles/profiles.html>

[17] Library of Congress gateway access <http://www.loc.gov/z3950>

[18] Mercury Z39.50 Client Homesite <http://www.basedowinfosys.com/projects/mzc>

[19] BibAdmin project page <http://bibadmin.gforge.inria.fr>

[20] Biblioscape – bibliographic software homepage <http://www.biblioscape.com>

[21] Thomson EndNote homepage <http://www.endnote.com>

[22] Thomson WriteNote homepage <http://www.writenote.com>

[23] CiteSeer web katalog <http://citeseerx.ist.psu.edu>

[24] ACM portál <http://portal.acm.org>

[25] ProQuest RefWorks homepage <http://www.refworks.com>

[26] Aigaion project site <http://www.aigaion.nl>

[27] OHRT, Monte – ZMIEVSKI, Andrei, Smarty Manual, 2005 New Digital Group, Inc. URL: <<http://www.smarty.net/distributions/manual/en/Smarty-2.6.14-docs.pdf>>

[28] CONVISSOR Daniel, Pear DB Abstraction Layer documentation

URL: <<http://pear.php.net/package/DB>>

[29] Firefox Tidy Add-on project page <http://users.skynet.be/mgueury/mozilla/>

## 10 Přílohy

### 10.1 Uživatelská a instalační příručka

#### 10.1.1 Instalace

Aplikace potřebuje pro svoji funkci následující podmínky:

- Webový server s podporou SSL. Aplikaci jsem testoval na Apache 2.0.
- PHP 5.x
- PEAR framework, modul DB [28]
- MySQL 5.x
- Na serveru musí být nainstalován Ghostscript

Nasazení nové databáze publikací se oproti původní aplikaci komplikuje tím, že je potřeba spouštět externí programy, které konvertují text z dokumentů PDF a PS.

Kvůli dokumentům typu postscript musí být na serveru nainstalován program Ghostscript. Návod na instalaci naleznete na stránkách projektu [10]. Pro dokumenty PDF se používá program XPDF [9]. Distribuce pro x86 Linux je součástí aplikace (v adresáři /libs/xpdf), není třeba jej instalovat.

Webový server by měl podporovat protokol https. Informace o instalaci SSL na server Apache naleznete zde [3]. Nicméně funkční SSL není podmínkou, aby aplikace fungovala. SSL je potřeba zprovoznit jen z důvodu ochrany hesel při přihlašování uživatelů.

Aplikace je nakonfigurována tak, že výchozí nastavení nepočítá s podporou SSL. Pokud však máte nainstalovanou podporu SSL, lze aplikaci přikázat, aby ji používala. V souboru /common/common.inc.php editujte řádku, která definuje globální proměnnou `account\_url`. Název protokolu změňte z http na https. Uživatelé pak budou přesměrováni vždy na zabezpečený přihlašovací formulář.

Postupujte podle těchto kroků:

Napřed je nutné zkopírovat obsah adresáře implementation na pevný disk a změnit nastavení konfiguračních souborů. Změňte obsah konfiguračního souboru /common/db\_fns.inc.php. Naleznete v něm funkci db\_connect a v ní řádek:

```
$dsn = DB_TYPE . "://user:passwd@host/database_name" ;
```

Změňte DSN podle svých požadavků. Za database\_name dosadíte název databáze. Dále si všimněte souboru /install/dap.sql. Tento soubor je SQL skript, který obsahuje kompletní strukturu databáze, včetně dat, která jsem konvertoval z původní databáze publikací. Importujte tento SQL skript např. z příkazové řádky takto:

```
mysql -u [jméno uživatele] -p < dap.sql
```

Do adresáře /storage/ nakopírujte obsah adresáře /storage/ z původní aplikace.

Nainstalujte ghoscript, pokud ještě na serveru není nainstalován. Instrukce najdete na webu <http://pages.cs.wisc.edu/~ghost/>.

Přesvědčte se, že ./ghoscript/bin a ./ghoscript/lib jsou v proměnné PATH. Pokud ne, přidejte je.



Nyní můžete nakopírovat celý upravený obsah složky implementation do adresáře, kam má přístup webový server (bez adresáře install) a nastavit přístupová práva pro identitu uživatele se kterou vystupuje webový server, pro zápis do souborů a složek, kam je to zapotřebí. Na systémech typu UNIX nastavte příkazem chmod práva 0777 pro adresáře /log, /storage a /templates\_c. Ostatní adresáře by měly mít práva nastavena na 0755. Soubory v adresáři /lib/xpdf/x86-linux/ by měly mít nastavená práva 755 a všechny ostatní soubory by měly mít nastavená práva na 644. Zkontrolujte, zda jsou hodnoty přiřazeny správně.

Je-li na serveru zapnut safe mode, musíte otevřít soubor php.ini a nastavit proměnnou safe\_mode\_exec\_dir. Do této proměnné vložte plnou cestu k adresářům kde je uloženo xpdf (/.../lib/xpdf/x86-linux/) a ghostscript.

Spusťte pomocí webového prohlížeče skript ./data\_conversions/fdt.php. Poté adresář data\_conversion smažte.

Instalace je tím kompletní, můžete zkontrolovat funkčnost webovým prohlížečem. Přihlaste se do administračního rozhraní, přepněte na záložku „checked publications“, otevřete libovolnou publikaci a zkuste k ní přidat nějaký soubor. Neměla by se zobrazit žádná chybová hláška. Takto nastavená aplikace by měla být zcela funkční.

V databázi jsou vytvořeni 3 uživatelé:

outsider@outsider - minimální oprávnění

vodicm1@vodicm1 - zadavatel

fiserp@fiserp - administrátor

### 10.1.2 Přihlašování

Při prvním otevření aplikace budete vyzváni ke vložení uživatelského jména a hesla. Abyste mohli být přihlášení, váš prohlížeč musí podporovat cookies.

Pokud jste helso zapoměli, klikněte na odkaz „forgotten password“, poté vložíte do připraveného formuláře vaši emailovou adresu a kliknete na tlačítko „send request“. Na vaši emailovou adresu přijde potvrzující zpráva. Dále klikněte na odkaz, který vám přišel v potvrzující zprávě. Zobrazí se vám stránka s novým heslem, které vám aplikace sama přidělila. Heslo si poznamenejte a přihlašte se s ním do aplikace.

The screenshot shows a web interface with a light blue header bar. On the left of the header is the text 'Main page' and on the right is a 'log in' link. Below the header, there is a login form. It consists of two input fields: the first is labeled 'Login:' and the second is labeled 'Password:'. Below these fields is a button labeled 'log in'. At the bottom of the form area, there is a link labeled 'Forgotten password'.

**Obrázek 18 - přihlašovací formulář**

Po přihlášení se vám zobrazí hlavní stránka. V pravém horním rohu stránky je odkaz, kterým se dostanete na formulář správy vašeho uživatelského účtu. Zde si můžete vaše

heslo libovolně změnit.

[Main page](#) [Create new publication](#) [Import publication](#) [Admin interface](#) [Logged in as fiserp | log out](#)

[Personal profile](#) [Annotations](#)

**Login:** fiserp  
**Name:** Petr Fiser  
**Email:** fiserp@fel.cvut.cz  
**User role:** administrator

Current password:   
New password:   
Pass. repetition:

**Obrázek 19 - Uživatelský účet**

V uživatelském účtu lze zobrazit seznam vlastních poznámek. Zobrazuje se typ poznámky (osobní, globální), text poznámky a odkaz na publikaci, ke které se poznámka vztahuje.

[Main page](#) [Create new publication](#) [Import publication](#) [Admin interface](#) [Logged in as fiserp | log out](#)

[Personal profile](#) [Annotations](#)

Private note: *Review this article tomorrow.* GLFSR-A New Test Pattern Generator for Built-in-Self-Test

Private note: *Send this article to Martin.* Deterministic Built in Self Test

**Obrázek 20 - Poznámky k publikacím**

### 10.1.3 Rozhraní čtenářů

Rozhraní čtenářů umožňuje vyhledávat a zobrazovat publikace, které jsou uloženy v databázi. V horní části stránky je navigační menu, pomocí kterého můžete přejít na hlavní stranu, formuláře pro import nebo ruční zadání publikace a administrační rozhraní, pokud máte dostatečná oprávnění. V pravém horním rohu jsou odkazy, které vedou na formulář správy uživatelského účtu a odhlášení.

Na hlavní stránce se vám zobrazí vyhledávací formulář. Můžete vybírat, zda chcete vyhledávat podle názvů publikací a zároveň ve fulltextu dokumentů (záložka documents) nebo podle jmen autorů (záložka authors).

Dále vyberte v jakých kategoriích se bude vyhledávat a specifikujte, zda hledané publikace mají patřit alespoň do jedné z vybraných kategorií (search operator or) nebo zda mají být obsaženy ve všech vybraných kategoriích (operátor and).

Pokud si nepřejete výsledky filtrovat dle kategorií, klikněte na odkaz „disable search in categories“; okno s kategoriemi se zavře a filtrování dle kategorií bude vypnuté.

The screenshot shows a web interface for searching documents. At the top, there are tabs for 'Documents' and 'Authors'. Below the tabs is a search input field containing the text 'circuits' and a 'search' button. Below the search bar, there is a section titled 'Disable search in categories' which is currently expanded. Inside this section, there is a 'Publication categories' list with checkboxes for various categories: ATPG, BDD, BIST, Cellular automata, Covering problem, Decomposition, Diagnostika, EXOR, Synthesis, FPGA, hardware, Logic design, On-line testing, Orthogonalization, SAT, Symmetric functions, and Verification. The 'Search operator' is set to 'OR'. At the bottom of the categories list, there is a link 'check/uncheck all'.

**Obrázek 21 - Vyhledávací formulář**

#### *Vyhledávání jen dle kategorií:*

Často se vám pravděpodobně bude hodit pouze vyhledávání jen podle kategorií. Budete-li chtít zobrazit všechny publikace ve vybraných kategoriích, jednoduše nevyplňujte textové vyhledávací pole. Pouze zaškrtněte kategorie a klikněte na tlačítko „search“.

#### *Vyhledávání v dokumentech:*

Do vstupního pole vložte frázi, kterou chcete hledat. Hledání se aplikuje na názvy publikací a fulltext přiložených dokumentů. Nezáleží na velikosti písmen ani diakritice. Necháte-li pole frází prázdné, hledat se bude bez ohledu na název publikace.

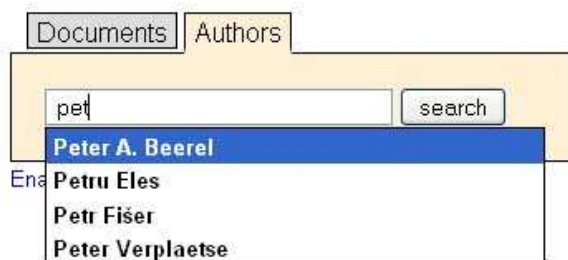
K vyhledávání můžete používat tyto operátory:

+	Každý nalezený záznam musí obsahovat toto slovo.
-	Žádný nalezený záznam nesmí obsahovat toto slovo.
( )	Závorky spojují slova do subvýrazu. Subvýrazy se mohou vnořovat.
~	Způsobí, že slovo bude přispívat k celkové relevanci negativně. Funguje podobně jako operátor „-“, ale nevyloučí záznam z množiny výsledků úplně.
*	Jako jediný operátor se dává na konec slova. Slouží jako náhražka libovolných znaků..
“	Výsledkem vyhledávání výrazu uzavřeného dvojími uvozovkami jsou záznamy, které obsahují přesně takovýto výraz.

Chcete-li najít publikace, které obsahují slovo „jablko“ a neobsahují slovo „hruška“, jednoduše zadejte výraz „+jablko –hruška“.



### Vyhledávání podle jmen autorů:

Přepněte vyhledávací formulář do režimu vyhledávání autorů. Zadejte jméno nebo více jmen autorů. Shodují-li se byť jen částečně s autory, kteří jsou uloženi v databázi, zobrazí se vám interaktivní nabídka. Viz následující obrázek:

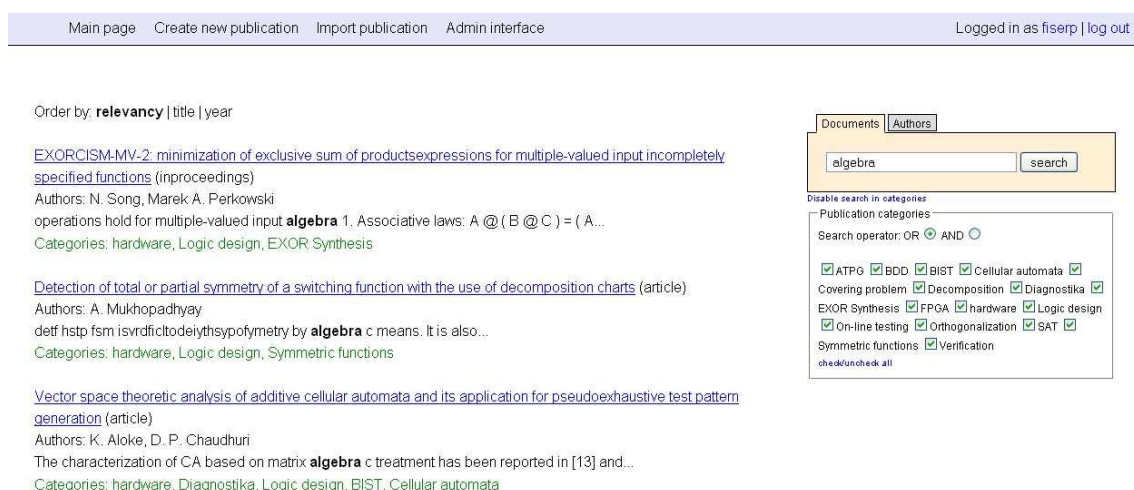


Obrázek 22 - Našeptávač se jmény autorů

### Výsledky hledání:

Stiskem tlačítka „search“ se dostanete na stránku s nalezenými publikacemi. Publikace můžete řadit dle relevance (výchozí), názvu a roku vydání. Výsledky obsahují také úryvky nalezeného textu a v něm tučně zvýrazněné hledané výrazy a seznam kategorií, ve kterých je publikace zařazena. Jsou-li k publikaci přiřazeny nějaké uživatelské poznámky, zobrazí se u jejího názvu značka  {zelená pokud jsou to cizí globální poznámky a  růžová, pokud jsou to vaše vlastní poznámky. Pokud na tyto značky najedete myší, zobrazí se vám vysvětlivky. Kliknutím na značku otevřete poznámky.

Je-li nalezených výsledků více než je povolená mez, zobrazí se vám na konci stránky navigační odkaz na další stránku s výsledky. Kliknutím na konkrétní publikaci zobrazíte její detail.



Obrázek 23 - Stránka s výsledky vyhledávání

### Detail publikace:

Detail publikace má 3 záložky. První záložka vám zobrazuje bližší informace o publikaci, BibTeX export, přiložené dokumenty ke stažení a citaci ve formátu, který si můžete vybrat z nabídky.

**EXORCISM-MV-2: minimization of exclusive sum of productexpressions for multiple-valued input incompletely specified functions**

Authors: N. Song, Marek A. Perkowski  
 Categories: hardware, Logic design, EXOR Synthesis

Overview   Modify   Annotations

**Type of publication:** inproceedings

**Booktitle:** The Twenty-Third International Symposium on Multiple-Valued Logic

**Authors:** N. Song, Marek A. Perkowski

**Pages:** 132-137

**Year:** 1993

**Location of conference:** Sacramento California USA

**ISBN:** 0-8186-3350-6

**Bibtex definition:**

```

@inproceedings{1993,
  author = "N. Song and Marek A. Perkowski",
  booktitle = "The Twenty-Third International Symposium on Multiple-Valued Logic",
  title = "EXORCISM-MV-2: minimization of exclusive sum of productexpressions for multiple-valued input incompletely specified functions",
  year = "1993",
  location = "Sacramento California USA",
  pages = "132-137"
}

```

**Downloads:**  
 00289569.pdf

Citation in ČSN ISO 690 ok  
 SONG, N. - PERKOWSKI, A. Marek. *EXORCISM-MV-2: minimization of exclusive sum of productexpressions for multiple-valued input incompletely specified functions.*; In: The Twenty-Third International Symposium on Multiple-Valued Logic (Sacramento California USA), 1993, ISBN 0-8186-3350-6, s. 132-137.

Obrázek 24 - Detail publikace

Druhá záložka vám nabízí možnost modifikovat publikaci. Modifikovat publikace však mohou jen uživatelé s patřičným oprávněním, takže pokud vám toto oprávnění chybí, na 2. záložku vám nebude umožněn přístup.

**EXORCISM-MV-2: minimization of exclusive sum of productexpressions for multiple-valued input incompletely specified functions**

Authors: N. Song, Marek A. Perkowski  
 Categories: hardware, Logic design, EXOR Synthesis

Overview   **Modify**   Annotations

Publication type\*: InProceedings (an article in a conference proceedings)

Title\*: EXORCISM-MV-2: minimization of exclusive sum of productexpressions for multiple-valued inp

Abstract:

Volume:

Number:

Chapter:

Pages:

Series:

Obrázek 25 - Modifikační formulář

Poslední záložka slouží k psaní poznámek k publikacím. Jsou-li k nějaké publikaci přiloženy poznámky pod vaší identitou, můžete takové poznámky dále editovat i mazat. Poznámky ostatních uživatelů nelze nijak upravovat nebo mazat. Jedině v případě máte-li administrační oprávnění.

## 10.1.4 Rozhraní pro zadavatele

### *Ruční zadání publikace:*

Rozhraní pro ruční zadávání nových publikací je přístupné pouze uživatelům s oprávněním „zadavatel“ a „administrátor“. Otevřete jej z menu pod položkou „Create new publication“.

Toto rozhraní se skládá celkem ze 3 formulářů. Na prvním musíte vyplnit typ a název publikace, volitelně můžete zadat i abstract a zařadit publikaci do některé z kategorií. Viz obrázek:

Main page   Create new publication   Import publication   Admin interface   Logged in as fiserp | log out

Type of publication\*:

Name of publication\*:

Abstract:

Place the publication into a category:

Categories

☐ ATPG ☐ BDD ☐ BIST ☐ Cellular automata ☐ Covering problem ☐ Decomposition ☐ Diagnostika ☐ EXOR  
☐ Synthesis ☐ FPGA ☐ hardware ☐ Logic design ☐ On-line testing ☐ Orthogonalization ☐ SAT ☐ Symmetric  
functions ☐ Verification

**Obrázek 26 - Rozhraní pro zadavatele**

Druhý formulář se odvíjí od typu publikace, který jste zadali v prvním formuláři. Položky označené hvězdičkou by měly být vyplněny. Pokud je nevyplníte, nic se nestane. Můžete pokračovat dál a na 3. formuláři budete pouze upozorněni na položky, které měly být vyplněny pro správnou kompatibilitu se systémem BibTeX, ale chybí.

Na 2. formuláři publikaci přiřazujete autory a vydavatele. Pokud je publikace článek, vybíráte časopis, do kterého článek patří. U publikace typu stat' v elektronickém sborníku (InProceedings) a sborník (Proceedings) vybíráte konferenci. Viz obrázek č. 26. Překlikáváním mezi konferencemi se spouští automatický mechanismus, který stahuje ze serveru details k jednotlivým konferencím a plní jimi formulář. Tento stav je vidět na obrázku 27. Aby tato funkcionality byla přístupná, musí váš prohlížeč podporovat javascript

Kliknutím na žluté pole „... abort editing“ uvedete formulář zpět do původního stavu, kde místo editace můžete přidávat nové konference.

Výběr autorů je znázorněn obrázkem 28. Pomocí tlačítek s šipkami přesouváte autory ze seznamu nalevo mezi vybrané autory a obráceně.



Name\*:

Abbreviation:

Year:  From:  To:

Location:

ISBN:

11th IEEE Design and Diagnostics of Electronic Circuits and Systems Workshop 2008 (2008)  
11th IEEE/ACM International Workshop on Logic & Synthesis (2002)  
18th IEEE international Symposium on Defect and Fault Tolerance in VLSI Systems (2003)  
21st International Conference on Computer Design (2003)  
26th Conference on Design Automation (1989)  
27th ACM/IEEE conference on Design automation (1991)  
29th ACM/IEEE conference on Design automation (1992)  
29th ACM/IEEE Design Automation Conference (1992)  
30th international conference on Design automation (1993)  
32nd ACM/IEEE conference on Design automation (1995)

Obrázek 27 - Číselník konference

**You can edit the conference. Click here to abort editing.**

Name\*:

Abbreviation:

Year:  From:  To:

Location:

ISBN:

11th IEEE Design and Diagnostics of Electronic Circuits and Systems Workshop 2008 (2008)  
11th IEEE/ACM International Workshop on Logic & Synthesis (2002)  
18th IEEE international Symposium on Defect and Fault Tolerance in VLSI Systems (2003)  
21st International Conference on Computer Design (2003)  
26th Conference on Design Automation (1989)  
27th ACM/IEEE conference on Design automation (1991)  
29th ACM/IEEE conference on Design automation (1992)  
29th ACM/IEEE Design Automation Conference (1992)  
30th international conference on Design automation (1993)  
32nd ACM/IEEE conference on Design automation (1995)

Obrázek 28 - Číselník konference v režimu update

name	middle	surname	
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="button" value="add author"/>

V. K. Agarwal  
V. D. Agrawal  
K. Alope  
Adnan Aziz  
Felice Balarin  
Luis Basto  
John Bean  
Peter A. Beerel  
Michael Belch  
Bart Benne  
Per Bjesse  
B. Bollig  
Arne Borälv  
Robert K. Brayton  
Melvin A. Breuer  
Franc Brglez  
Franca Brglez  
D. Bryan  
Randal E. Bryant  
Jerry R. Burch  
Jon T. Butler  
Adam Cabe  
Roberto Wolfier Calvo  
Jan Van Camphenout

>> Michael Belch  
<< Franc Brglez  
Jerry R. Burch

Obrázek 29 - Formulář pro výběr / vložení autorů

Na třetím posledním formuláři se vám zobrazí souhrn dat, která jste zadali, případně validační report systému BibTeX. Na tomto formuláři máte možnost vložit k publikaci souborové přílohy. Kliknete-li na vstupní pole pro zadání souboru, automaticky se vám vytvoří další, abyste mohli vložit další soubor. Viz obrázek 29.

Main page   Create new publication   Import publication   Admin interface   Logged in as fiserp | log out

Publication you are inserting has particular defects, which you may ignore and save the publication. For the compatibility with BibTeX format we recommend go back to the previous form and correct these defects:

- Year is missing

**Type of publication:** book  
**Title:** Design Pattern Discovery  
**Publisher:** IEEE Computer Society  
**Address:** Washington, DC, USA  
**Authors:** Michael Belch, Franc Brglez, Jerry R. Burch

You can add files to the publication (pdf, ps, ppt, txt, doc, odt, zip, rar):

Procházet...

<< previous   finish

**Obrázek 30 - Přehled zadaných dat a validační report**

#### *Import publikace z definice BibTeX:*

Rozhraní import BibTeX definic je přístupné pouze uživatelům s oprávněním „zadavatel“ a „administrátor“. Otevřete jej z menu pod položkou „Import publication“. Pomocí něj můžete importovat do databáze libovolnou publikaci, máte-li její definici. Některé katalogy jako např. CiteSeer [23] a ACM [24] nabízejí tyto definice ke stažení.

Vložte definici do textového pole „BibTeX definition“ a klikněte na tlačítko „next“. Proběhne-li import v pořádku, zobrazí se vám přehled importovaných dat, respektive validační report jako při ručním zadávání publikace viz obr. 29. V opačném případě, je-li v definici syntaktická chyba; zobrazí se vám hláška s popisem chyby a číslem řádky, na které se vyskytla. Následující obrázek ukazuje definici se syntaktickou chybou, kterou jsem stáhl z portálu ACM. ACM portál na konec definice přidává chybně čárku.



Line #12: Unexpected token RPAR . Expected KEYWORD.

Insert BibTeX definition:

```
@inproceedings{1142436,
  author = {T. Scott Saponas and Madhu K. Prabhaker and Gregory D. Abowd},
  title = {The impact of pre-patterns on the design of digital home applications},
  booktitle = {DIS '06: Proceedings of the 6th conference on Designing Interactive
systems},
  year = {2006},
  isbn = {1-59593-367-0},
  pages = {189--198},
  location = {University Park, PA, USA},
  doi = {http://doi.acm.org/10.1145/1142405.1142436},
  publisher = {ACM},
  address = {New York, NY, USA},
}
```

Place the publication into a category:

Categories

☐ ATPG ☐ BDD ☐ BIST ☐ Cellular automata ☐ Covering problem ☐ Decomposition ☐ Diagnostika ☐ EXOR Synthesis ☐

**Obrázek 31 - Import definice se syntaktickou chybou**

Proběhne-li syntaktická analýza v pořádku, zobrazí se vám formulář, který zhodnotí kvalitu BibTeXové definice (zda obsahuje povinná políčka). Pokud je něco v nepořádku zobrazí pouze krátký report. Nyní můžete k publikaci přiložit libovolné dokumenty a po stisknutí tlačítka „finish import“ se publikace vloží do databáze.

### 10.1.5 Administrační rozhraní

Administrační rozhraní se skládá ze 10 přepínatelných záložek, ve kterých jako administrátor vybíráte zvolenou funkcionalitu. Musí být zapnuté cookies. V případě vypnutých cookies, nebude úplně korektně fungovat formulář pro editaci publikace v záložce 1 a 2.

Zde je uveden význam jednotlivých záložek:

1. schvalování, editace a mazání nezkontrolovaných publikací
2. editace a mazání zkontrolovaných publikací
3. editace, mazání a přidávání rozšířených atributů, editace jejich viditelnosti (veřejný/neveřejný)
4. editace, mazání a přidávání kategorií
5. editace, mazání a přidávání autorů
6. editace, mazání a přidávání vydavatelů
7. editace, mazání a přidávání časopisů
8. editace, mazání a přidávání konferencí
9. editace šablon, které určují formát citace
10. správa uživatelů

V první záložce je okamžitě vidět seznam publikací, které byly nově zadány nebo starších publikací, které někdo editoval. Po kliknutí na konkrétní položku se vám

zobrazí stručný náhled na vlastnosti publikace. Jsou zobrazeny i odkazy na přiložené soubory s možností přidat další a nebo vymazat všechny. Jestliže zadavatel přiložil k publikaci rozšířený atribut, budete vidět i ten. Vy máte dále na výběr několik možností co s publikací provést.



**Obrázek 32 - administrační rozhraní**

#### *Upravit publikaci:*

Této volby využijete v případě, že chcete publikaci poopravit nebo nějak doplnit. Zpřístupní se vám kliknutím na tlačítko „upravit“. Zobrazíte tím formulář připomínající zadavatelské rozhraní. Pracuje se s ním velice podobně s tím rozdílem, že není dělen do separátních částí, ale zobrazuje přehledně všechny vlastnosti najednou a pole, která nemají být vyplněna (což je pevně dáno typem publikace) se nezobrazují. Změnou typu publikace se opět okamžitě zneplatní položky, které typu publikace nepřísluší. Je nutno podotknout, že aby tato funkce fungovala musí být zapnut javascript. V opačném případě budou přístupná všechna políčka, což ale není žádná tragédie, pouze se z formuláře vytratí jeho elegance. O krok zpět na předchozí obrazovku se můžete vrátit kliknutím na znak „<<“ v levém horním rohu záložky.

#### *Označit publikaci jako schválenou:*

Shlédl-li jste novou publikaci a chcete-li ji označit jako zkontrolovanou, klikněte na tlačítko „mark as checked“. Publikace se přesune mezi zkontrolované, o čemž se okamžitě můžete přesvědčit.

#### *Smazat publikaci:*

Kliknutím na tlačítko „smazat“, úplně odstraníte publikaci z databáze.

Záložky pro rozšířené atributy, kategorie, autory, časopisy a konference fungují všechny vcelku na podobném principu. Zobrazují veškeré záznamy, např. pátá záložka vypisuje jména autorů. Ti se dají řadit kliknutím na horní modrý popisek. Řadí vzestupně podle kritéria, které vyberete. Položky je možné editovat stisknutím tlačítka „modify“.

Kliknutím na tlačítko „smazat“, položku trvale odstraníte z databáze a všechny vazby mezi touto položkou a publikacemi, které se na ní odkazovaly budou zrušeny. Tato operace je svým způsobem nevratná, takže je zapotřebí se rozmyslet, zda je opravdu důvod položku mazat (většinou lze vystačit pouze se samotnou editací). O tom, zda můžete položky bez obav mazat nebo ne, vás informuje varovná značka.

[Main page](#)
[Create new publication](#)
[Import publication](#)
[Admin interface](#)

Logged in as fiserp | [log out](#)

[Waiting for check](#)
[Checked](#)
[Attributes](#)
[Categories](#)
[Authors](#)
[Publishers](#)
[Journals](#)
[Conferences](#)
[Templates](#)
[Users](#)

name	middlename	surname			
V.	K.	Agarwal	<a href="#">change</a>	<a href="#">delete</a>	
V.	D.	Agrawal	<a href="#">change</a>	<a href="#">delete</a>	
K.		Aloke	<a href="#">change</a>	<a href="#">delete</a>	
Adnan		Aziz	<a href="#">change</a>	<a href="#">delete</a>	
Felice		Balarin	<a href="#">change</a>	<a href="#">delete</a>	
Luis		Basto	<a href="#">change</a>	<a href="#">delete</a>	
John		Bean	<a href="#">change</a>	<a href="#">delete</a>	

**Obrázek 33 - správa autorů**

#### *Editace šablon:*

Další možností, kterou jako administrátor disponujete, je editace šablon. Tato funkce je vám přístupná v deváté záložce pod názvem „templates“. Tlačítkem save po předchozí editaci šablon, změníte šablony, do kterých systém dosazuje texty citací ve čtenářském rozhraní. Ke správnému pochopení syntaxe šablon prostudujte dokument [27].

#### *Práce s rozšířenými atributy:*

Ve 3. záložce pod názvem „attributes“ můžete manipulovat s rozšířenými atributy. Dle vašeho uvážení můžete přidávat publikacím další vlastnosti. Když pak budete chtít, aby se tyto atributy zobrazovaly v citacích, upravte šablony v 9. záložce (například prostým přidáním elementu {\$\_mujatribut}).

## 10.2 Obsah příloženého CD

/doc	adresář s dokumentací včetně tohoto textu
/implementation	zdrojové skripty
/BP	adresář s původní bakalářskou prací

### *Obsah adresáře /implementation:*

./account/	skripty ke správě uživatelského účtu
./admin/	soubory pro administračního rozhraní
./common/	sdílené soubory, knihovny funkcí, tříd atd.
./common/pear	soubory Pear frameworku
./graphics/	obrázky a ikony
./import/	skripty rozhraní importu
./install/	obsahuje soubory potřebné k instalaci
./libs/	soubory knihovny Smarty, AjaxAC a XPdf
./log/	soubory logů
./load/	soubory rozhraní pro zadávání nových publikací
./storage/	adresář souborového úložiště
./templates/	šablony
./templates_c/	kompilované šablony, automaticky generované třídou Smarty

### *Popis součástí projektu v adresáři /implementation:*

./detail_provider.php	slouží jako provider Ajax požadavků na detail číselníku
./document_provider.php	slouží jako poskytovatel dokumentů
./index.php	vstupní bod čtenářského rozhraní
./search_bar.css	styly vyhledávacího formuláře
./search_bar.js	klientský skript vyhledávacího formuláře
./suggest.js	klientský skript našeptávače
./Suggest.inc.php	serverový handler našeptávače
./User.inc.php	součástí je třída User, reprezentuje čtenářské rozhraní
./user.css	styly pro čtenářské rozhraní
./admin/Admin.inc.php	abstraktní třída Admin
./admin/AdminT12.inc.php	finální třída AdminT12
./admin/AdminT3.inc.php	finální třída AdminT3
./admin/AdminT4.inc.php	finální třída AdminT4
./admin/AdminT5.inc.php	finální třída AdminT5
./admin/AdminT6.inc.php	finální třída AdminT6

./admin/AdminT7.inc.php	finální třída AdminT7
./admin/log_info.php	zpráva o odhlášení
./admin/logout.php	odhlašovací skript
./admin/admin.css	soubor stylů pro administrační rozhraní
./admin/index.php	vstupní bod administračního rozhraní
./common/common.inc.php	soubor globálních definic a konstant
./common/db_fns.inc.php	databázové funkce
./common/misc_fns.inc.php	specializované funkce
./common/Parser.inc.php	třída Parser
./common/SException.inc.php	třída SException
./common/SAException.inc.php	třída SAException
./load/index.php	vstupní bod části aplikace určené pro zadávání publikací
./load/load.css	soubor stylů pro zadavatelské rozhraní
./load/Load.inc.php	třída Load
./templates/admin_header.tpl	záhlaví zadavatelské části
./templates/admin_header.tpl	záhlaví administrační části
./templates/admin_tab12.tpl	šablona 1. a 2. záložky administračního rozhraní
./templates/admin_tab3.tpl	šablona 3. záložky administračního rozhraní
./templates/admin_tab4.tpl	šablona 4. záložky administračního rozhraní
./templates/admin_tab5.tpl	šablona 5. záložky administračního rozhraní
./templates/admin_tab6.tpl	šablona 6. záložky administračního rozhraní
./templates/admin_tab7.tpl	šablona 7. záložky administračního rozhraní
./templates/admin_tab8.tpl	šablona 8. záložky administračního rozhraní
./templates/admin_tab9.tpl	šablona 9. záložky administračního rozhraní
./templates/admin_tab10.tpl	šablona 10. záložky administračního rozhraní
./templates/load_st1.tpl	šablona 1. formuláře zadavatelského rozhraní
./templates/load_st2.tpl	šablona 2. formuláře zadavatelského rozhraní
./templates/load_st3.tpl	šablona 3. formuláře zadavatelského rozhraní
./templates/load_stFinish.tpl	šablona konečného formuláře zadavatelského rozhraní
./templates/user.tpl	šablona uživatelského rozhraní

### 10.3 Rozdělení publikací a jejích atributů v systému BibTeX

<i>Pole podporovaná publikacemi v BibTeXu</i>													
	article	book	booklet	inbook	incollection	inproceedings	manual	masterthesis	misc	phdthesis	proceedings	techreport	unpublished
address		o	o	o	o	o	o	o		o	o	o	
author	m	p	o	p1	m	m	o	m	o	m		m	m
booktitle					m	m							
chapter				p2	o	o							
edition		o		o	o		o						
editor		p		p1	o	o			o		o		
howpublished			o						o				
institution												m	
journal	m												
month	o	o	o	o	o	o	o	o	o	o	o	o	o
note	o	o	o	o	o	o	o	o	o	o	o	o	m
number	o	o		o	o	o					o	o	
organization						o	o				o		
pages	o	o	o	p2	o	o	o	o	o	o	o	o	o
publisher		m		m	m	o					o		
school								m		m			
series		o		o	o	o					o		
title	m	m	m	m	m	m	m	m	m	m	m	m	m
type												o	
volume	o	o		o	o	o					o		
year	m	m	o	m	m	m	o	m	o	m	m	m	o

**Význam zkratek:** m – mandatory, o – obligatory, p – partly(vyjadřuje povinnost jednoho nebo 2. pole)