VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

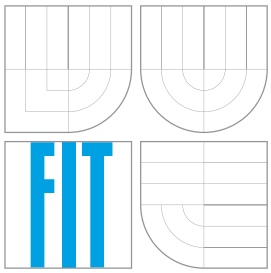
# ŘÍDICÍ SYSTÉM PROJEKTU RERESEARCH

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE IVO DLOUHÝ
AUTHOR

BRNO 2010

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÉ GRAFIKY A MULTIMÉDIÍ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER GRAPHICS AND MULTIMEDIA

# ŘÍDICÍ SYSTÉM PROJEKTU RERESEARCH

RERESEARCH MANAGEMENT SYSTEM

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

IVO DLOUHÝ

VEDOUCÍ PRÁCE
SUPERVISOR

Ing. LUBOMÍR OTRUSINA

BRNO 2010

**Vysoké učení technické v Brně - Fakulta informačních technologií**

Ústav počítačové grafiky a multimédií Akademický rok 2009/2010

# Zadání bakalářské práce

Řešitel: **Dlouhý Ivo**
Obor: Informační technologie
Téma: **Řídicí systém projektu ReReSearch**
**ReReSearch Management System**
Kategorie: Web

Pokyny:

1. Seznamte se s výstupy systému CiteSeer a velkých digitálních knihoven (ACM, IEEE, ...).
2. Navrhněte systém řízení a aktualizace databáze systému ReReSearch.
3. Implementujte navržený systém s důrazem na efektivitu.
4. Připravte demonstrační data, která dovolí zhodnotit výkonnosti systému.
5. Vytvořte stručný plakát prezentující práci, její cíle a výsledky.

Literatura:

- podle dohody

Při obhajobě semestrální části projektu je požadováno:

- funkční prototyp řešení

Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese http://www.fit.vutbr.cz/info/szz/

Technická zpráva bakalářské práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap (20 až 30% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Otrusina Lubomír, Ing.**, UPGM FIT VUT
Datum zadání: 1. listopadu 2009
Datum odevzdání: 19. května 2010

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav počítačové grafiky a multimédií
612 66 Brno, Božetěchova 2

L.S.

doc. Dr. Ing. Jan Černocký
*vedoucí ústavu*

# Abstrakt

Práce se zabývá návrhem a implementací databáze, archivu souborů, systému aktualizace a rozhraní pro komunikaci modulů v rámci projektu ReReSearch. Obsahuje analýzu digitálních knihoven a systému ReReSearch, vysvětluje pojmy a dále popisuje návrh, implementaci a testování zmíněných součástí.



# Abstract

Aim of this thesis is to design and implement database, file archive, webpage change monitoring system and module communication API in ReReSearch project. It also contains analysis of digital libraries and ReReSearch system, explains terms and describes design, implementation and testing of these components.



# Klíčová slova

digitální knihovna, databáze, archiv souborů, sledování změn stránek, XML rozhraní

# Keywords

digital library, database, file archive, webpage change monitoring, XML API


# Citace

Ivo Dlouhý: Řídicí systém projektu ReReSearch, bakalářská práce, Brno, FIT VUT v Brně, 2010

# Řídicí systém projektu ReReSearch

## Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením Ing. Lubomíra Otrusiny. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Ivo Dlouhý
19. května 2010

## Poděkování

Rád bych poděkoval Ing. Lubomíru Otrusinovi za vstřícný přístup, hodnotné rady a odborné vedení mé práce.

# Obsah

# Seznam obrázků

# Kapitola 1

# Úvod

Digitální knihovny jsou pojem, který slyšíme s rozmachem informačních technologií stále častěji. Umožňují vytváření elektronických sbírek dokumentů jednoduše přístupných uživatelům. Po spojení infrastruktury digitálních knihoven s moderními metodami extrakce informací získáváme citační databázi, užitečný nástroj pro získávání informací.

Bakalářská práce je součástí projektu ReReSearch a proto je vhodné přesně vymezit autory některých zmiňovaných součástí. Původní verze řídicího systému pochází z bakalářské práce Jana Horáka. Jak bude uvedeno dále, každý z modulů má svého autora, případně autory. Schéma databáze je založeno na původním schématu navrženém Peterem Bartošem a Michalem Angelovem.

Práce je orientována na digitální knihovny a citační databáze zabývající se vědeckými publikacemi. Přispívá k řešení projektu ReReSearch, který si klade za cíl vytváření webových portálů umožňujících budování báze znalostí. Cílem bakalářské práce je analýza, návrh centralizované databáze, aktualizace webových zdrojů a rozhraní pro komunikaci modulů.

V kapitole 2 se budeme zabývat analýzou problému. Vymezíme pojem digitální knihovna, popíšeme významné zástupce. Dále uvedeme cíle a plánované použití systému ReReSearch v rámci kterého bude práce realizována. Seznámíme se také s moduly projektu ReReSearch, jejich úkolem, vstupy a výstupy. Na základě předchozí analýzy a seznámení s řídícím systémem navrhneme cíle práce. Kapitola 3 obsahuje stručné uvedení do problematiky pomocí výkladu několika důležitých pojmů a technologí. Do kapitoly 4 je zařazen návrh řešení cílů stanovených v kapitole 2 za pomoci technologií z kapitoly 3. Jsou zde uvedeny postupy použité pro tvorbu databáze, archivu souborů a metody použité pro vyhledání změn ve webových stránkách. Kapitola 5 popisuje implementaci vybraných postupů navržených v kapitole 4, zejména nástroje pracující s databází a XML rozhraní určené pro komunikaci modulů. Dále je zde vysvětlena adresářová struktura řídícího systému. Kapitola 6 se zabývá hodnocením implementovaných systémů, vytvořením demonstračních dat a stanovením časových nároků pro vybrané akce systému ReReSearch.

Jelikož se jedná o dlouhodobější vývoj systému, některé informace nacházející se v této bakalářské práci byly autorem publikovány v interní dokumentaci projektu, na kterou se lze také odkázat pro podrobnější informace.

# Kapitola 2

# Analýza problému

Pro vývoj řídícího systému rozsáhlého projektu, který se skládá z většího množství různorodých modulů, je nezbytné analyzovat problém z více pohledů. Nejdříve je třeba shromáždit informace, zejména účel projektu, důkladnou analýzu stavu a funkčnosti součástí, ale také údaje o podobných projektech a používaných trendech nebo technologiích. Na základě analýzy lze stanovit cíle práce.

## 2.1 Digitální knihovny

Digitální knihovna je často používaný pojem, který charakterizuje systémy se stejným účelem, ale mnohdy velmi rozdílnou funkcionalitou. Označujeme tak systémy od skladišť dat a elektronického obsahu, přes citační databáze po systémy pro správu obsahu, ale i komplexní systémy vyvíjené ve vědeckém prostředí.

Často je této v souvislosti uváděna tato definice: Digitální knihovna je spravovaná sbírka informací spolu s odpovídajícími službami, přičemž informace jsou uloženy v digitální podobě a jsou dostupné prostřednictvím sítě [1].

Pojmem digitální knihovna se označují často odlišné systémy, přesto lze najít několik společných znaků [2]:

- Základní funkcí je organizace elektronické sbírky, nikoliv digitalizace fyzického materiálu.
- Informační zdroje tvořící digitální knihovnu můžeme označit jako heterogenní (způsobem uložení/organizací/správou objektů a použitými platformami), dynamické (začleňováním a vyřazováním komponent) a multimediální (povahou dat).
- Digitální knihovna je realizována více informačními komponentami, které musí být vhodně propojeny.
- Úkolem digitální knihovny je zajistit uživateli jednotný přístup k relevantním digitálním informacím bez ohledu na jejich formu, formát, způsob a místo uložení.

Uvedené znaky vhodně shrnuje referenční model [3], zpracovaný sdružením pro digitální knihovny DELOS[1].

Pojem digitální knihovna charakterizuje stále se vyvíjející organizaci, která vzniká spojením několika komponent. Referenční model rozlišuje tři součásti digitální knihovny:

[1] http://www.delos.info/

- **Digitální knihovna:** Organizace, která sbírá, spravuje a dlouhodobě archivuje digitální obsah. Komunitě uživatelů nabízí funkcionalitu měřitelné kvality podle předem daných zásad.
- **Systém digitální knihovny:** Softwarový systém založený na definované, často distribuované architektuře, poskytující funkcionalitu potřebnou pro digitální knihovnu. Uživatelé komunikují s digitální knihovnou prostřednictvím systému digitální knihovny.
- **Řídicí systém digitální knihovny:** Obecný softwarový systém poskytující nezbytnou softwarovou infrastrukturu potřebnou pro vytvoření a správu systému digitální knihovny. Zajišťuje základní funkce digitální knihovny a umožňuje začlenění dalšího software poskytujícího přidanou funkcionalitu.

Jedním z typů řídicího systému digitální knihovny je systém digitální knihovny skladiště [3]. Tento typ je specifický tím, že je tvořen komponentami, které zapouzdřují klíčovou funkcionalitu digitální knihovny, a nástroji, pomocí kterých je možné komponenty propojovat pro vytvoření systému digitální knihovny.

Jako příklady digitálních knihoven je možno uvést následující projekty. Vzhledem k zaměření projektu ReReSearch, které je popsáno v kapitole 2.2, budou dále zmiňovány pouze služby, které se zabývají vědeckými publikacemi.

### 2.1.1 Google Scholar

Služba Google Scholar[2] poskytuje centralizované vyhledávání vědecké literatury z různých zdrojů, od akademických vydavatelů až po profesionální společnosti. Výsledky vyhledávání jsou řazeny podle relevance, která je určena na základě textu publikace, autora, zdroje a míry citovanosti [4, 5].

Výhodami služby je rozsáhlá databáze pro vyhledávání, která má velký potenciál k rozšiřování díky spolupráci s vydavateli a knihovníky. Databáze obsahuje publikace všech zaměření, a proto zde můžeme najít také publikace zabývající se tématy z více oborů [6]. Google Scholar je propojen s databází digitalizovaných publikací spadajících do projektu Google Books[3]. Nevýhodou je mnohdy nesprávná detekce metadat, zejména jmen autorů, a s tím spojené nepřesnosti při vyhledávání [6]. Podobný problém se však v určité míře vyskytuje u všech databází, které obsahují automaticky zpracovaná data, v tomto směru je tedy velký prostor pro zlepšení.

### 2.1.2 ACM

ACM[4] (Association for Computing Machinery) je akademická a vědecká společnost, která umožňuje spolupráci odborníkům v oblasti informačních technologií, organizuje mnoho konferencí a spravuje digitální knihovnu ACM Digital Library[5]. Ta obsahuje archiv sborníků, publikací s časopisů publikovaných v rámci ACM, ale také obsah vybraných archivů třetích stran. Pro vyhledání je dostupných více než 1 milion publikací a citací obsahujících více než 2 miliony stránek textu [7, 8].

[2] http://scholar.google.com/
[3] http://books.google.com/
[4] http://www.acm.org/
[5] http://portal.acm.org/

### 2.1.3 IEEE

IEEE [6] (Institute of Electrical and Electronics Engineers) je neziskové sdružení odborníků za účelem podpory technologického pokroku. Podobně jako ACM, i IEEE spravuje digitální knihovnu – IEEE Xplore Digital Library[7]. Knihovna nabízí především sborníky, časopisy a standardy, publikované sdružením IEEE. Obsahuje více než 2,5 milionu publikací [9, 10].

### 2.1.4 CiteSeer

Známým představitelem digitálních knihoven a citačních databází je systém CiteSeer, který obsahuje vědeckou literaturu z oblasti informačních technologií. Cílem projektu je zjednodušit přístup k vědecké literatuře a zároveň poskytnout zdroje, algoritmy, data a technické informace pro provoz citační databáze, proto je také příkladem mnoha systémů digitálních knihoven a portálů. Byl prvním projektem, který poskytuje automatické indexování a propojování citací [11].

Systém CiteSeer byl původně vyvinut jako prototyp, ale během svého desetiletého provozu se setkal s velkým ohlasem a proto autoři připravují nástupce v podobě CiteSeer$^X$[8], v době psaní práce je k dispozici veřejná beta verze [12]. Databáze systému CiteSeer$^X$ obsahuje 1,6 milionu publikací provázaných 30 miliony citací a je volně dostupná ke stažení.

### 2.1.5 DBLP

Další z digitálních knihoven je DBLP[9] (Digital Bibliography & Library Project). Knihovny Google Scholar a CiteSeer nabízí rozsáhlou databázi díky strojovému zpracování dat. Oproti tomu databáze DBLP je udržována manuálně, znamená to tedy, že co do rozsáhlosti se nemůže měřit se zmíněnými systémy, nabízí však mnohem vyšší kvalitu záznamů [13]. Rovněž databáze DBLP je volně ke stažení.

### 2.1.6 Web of Knowledge

Velmi významným projektem je také ISI Web of Knowledge[10]. Služba je provozována společností Thomson Reuters a poskytuje jednotný přístup k databázím společnosti. Mezi databáze patří citační indexy vědeckých oborů, sociálních věd, umění a humanitních věd a chemický [14]. Jedná se o komerční projekt a přístup do databáze je umožněn pouze po zaplacení poplatku.

## 2.2 Projekt ReReSearch

Cílem projektu je vytvořit systém řídící tvorbu webových portálů, které umožňují budování báze poznatků ve vědeckém oboru [15].

Název ReReSearch charakterizuje proces orientace v novém oboru. Studenti, začínající vědci a výzkumníci, kteří přecházejí do nové oblasti, musí najít základní literaturu k danému tématu, relevantní publikace, renomované autory a vědecké kapacity s podobným zaměřením [16, 15]. Tento proces můžeme označit jako výzkum ve výzkumu, opakovaný výzkum – odtud název ReReSearch. Projekt může být znám také pod dřívějšími názvy PortaGe [16],

[6] http://www.ieee.org/
[7] http://ieeexplore.ieee.org/Xplore/guesthome.jsp
[8] http://citeseerx.ist.psu.edu/
[9] http://dblp.uni-trier.de/
[10] http://wok.mimas.ac.uk/

nebo Generátor vědeckých webových portálů [17], které vystihovaly základní cíl systému, tedy generování vědeckých webových portálů. Často je také používána zkratka RRS.

Dnes dostupné digitální knihovny disponují obsáhlou databází, vyhledávání informací však postrádá kontext. Nelze správně určit, které informace jsou pro uživatele relevantní. Oproti tomu, vědecké portály vytvářené systémem ReReSearch by vznikaly z počátečních dat nahraných uživatelem, na základě kterých by systém dokázal autonomně budovat bázi znalostí. V libovolném okamžiku by měl mít uživatel možnost do procesu vstoupit a upravit jej dle vlastních preferencí [16]. Na základě dostupných informací můžeme nastínit modelové funkce systému:

1. Uživatel inicializuje portál, nahraje oblíbené články, zadá zajímavé autory, stránky projektů nebo konferencí.

2. Systém na základě vstupů od uživatele začíná budovat bázi znalostí. Vyhledává nové autory a relevantní publikace.

3. Proces získávání znalostí může být plně řízen uživatelem. Každý uživatel má jiné potřeby, je tedy možné specifikovat typ a dostupnost publikací, obtížnost tématu, rychlost rozšiřování báze znalostí.

4. Na základě dostupných informací vyhodnotí systém, co je pro uživatele nejlepší a předloží mu výsledky.

## 2.3 Moduly

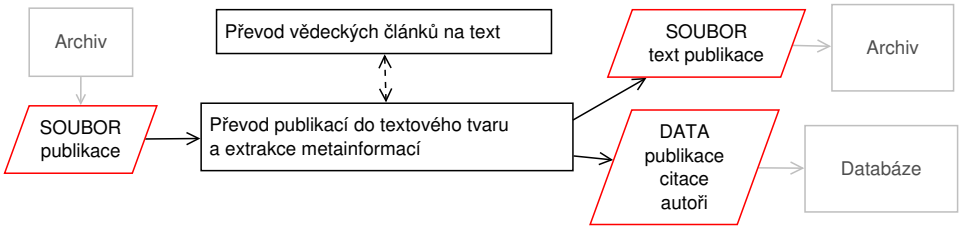
Funkcionalita systému ReReSearch je rozložena do mnoha modulů. Pro správný návrh databáze a souvisejících funkcí je nezbytné mít základní přehled o modulech, které tvoří systém.

U každého z dále uvedených modulů jsou uvedeni jeho autoři, princip fungování, vstupy a výstupy, které jsou zároveň přehledně znázorněny na připojených diagramech. Možná propojení modulů v rámci systému jsou uvedeny na celkovém diagramu v příloze B.

### 2.3.1 Převod publikací do textového tvaru a extrakce metainformací

Jak je zřejmé z názvu modulu, úkolem je zpracování publikací, konkrétně převedení elektronických verzí publikací do textové formy. V problematických případech je vhodné použít pokročilé technologie, které nabízí modul Převod vědeckých článků na text pracující se systémem OCR (Optical Character Recognition, optické rozpoznávaní znaků). Po převodu na text je z publikace extrahován název článku, jména autorů spolu s jejich adresami pro elektronické pošty, abstrakt, klíčová slova, názvy kapitol a také citace použité v publikaci, ze kterých jsou extrahovány všechny dostupné části. Pro extrakci je využito slovníků jmen, států a poté specifických metod pro každou z extrahovaných informací [18].

Výstupem modulu je textová forma publikace a extrahované metainformace. Textovou formu lze dále využít pro indexování a určení klíčových slov, získané metainformace pomáhají budovat databázi a mohou tak být využity například jako podnět pro vyhledání domovské stráky autora (modul je popsán v kapitole 2.3.6). Činnost modulu v rámci systému je graficky znázorněna na obrázku 2.1. Autorem modulu je Tomáš Lokaj, podmodul Převod problematických článků na text vytvořil Jiří Matička. Identifikátorem v řídicím systému je `publication_file_data`.

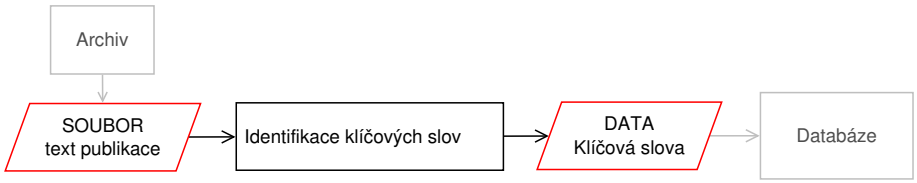



Obrázek 2.1: Modul Převod publikací do textového tvaru a extrakce metainformací

### 2.3.2 Identifikace klíčových slov

Textová podoba publikace může být využita k identifikaci klíčových slov. Dokument je nejdříve morfologicky a analyticky označkován za využití systému PDT[11] a TreeTagger[12]. Dále jsou z větných stromů vybrána klíčová slova, která jsou sjednocena pomocí thesauru. V textech v českém jazyce je nutné převést slova do správných rodů a pádů [19].

Vstupem je soubor s publikací v textové podobě, výstupem identifikovaná klíčová slova, která jsou použita k indexaci dokumentů (viz. 2.3.3). Modul je znázorněn na obrázku 2.2, autorem je Tomáš Strachota a identifikátorem v řídícím systému je `publication_file_keyword`.

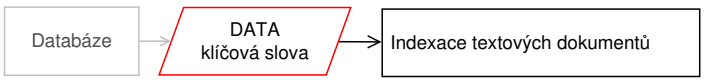



Obrázek 2.2: Modul Identifikace klíčových slov

### 2.3.3 Indexace textových souborů

Modul využívá klíčových slov z modulu Identifikace klíčových slov (viz. 2.3.2) spolu s textovou podobou publikace, ze kterých vytvoří index. Pro práci s indexy je využito nástroje Apache Solr [13].

Vstupem jsou klíčová slova, výstupem je soubor obsahující index, který je použit k vyhledávání sémanticky podobných dokumentů nebo termínů. Modul je znázorněn na obrázku 2.3, autorem je Erik Dresto a identifikátorem v řídícím systému je `publication_file_index`.



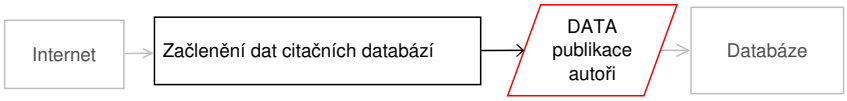

Obrázek 2.3: Modul Indexace textových souborů

[11] http://ufal.mff.cuni.cz/pdt/
[12] http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/
[13] http://lucene.apache.org/solr/

### 2.3.4 Začlenění dat citačních databází

Data v systému ReReSearch mohou být získána extrakcí z publikací a webových stránek, nebo také vložením kompletní sady dat z jiných systémů. Modul získá data z citačních databází, které podporují jejich šíření. Mezi takové databáze patří CiteSeerX (viz. 2.1.4), DBLP (viz. 2.1.5) a CSB[14] (The Collection of Computer Science Bibliographies) [20].

Výstupem modulu jsou získaná data, mezi která patří zejména publikace, jména autorů, místa vydání, citovanost, nebo také samotný soubor s článkem. Modul je znázorněn na obrázku 2.4, autorem je Martin Zelinka a identifikátorem v řídícím systému je `publication_import_data`.

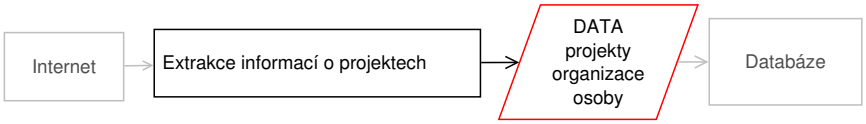



Obrázek 2.4: Modul Začlenění dat citačních databází

### 2.3.5 Extrakce informací o projektech

Dalším z modulů realizujících začlenění již existujících databází pracuje s projekty na portálech CORDIS[15] a EPSRC[16] (více informací lze získat v dokumentaci modulu) [21].

Výstupem modulu jsou informace o projektu a souvisejících organizacích a událostech. Údaje o projektu mohou být dále použity k získání výzkumných zpráv. Modul je znázorněn na obrázku 2.5, autorem je Ivan Homoliak a identifikátorem v řídícím systému je `project_import_data`.



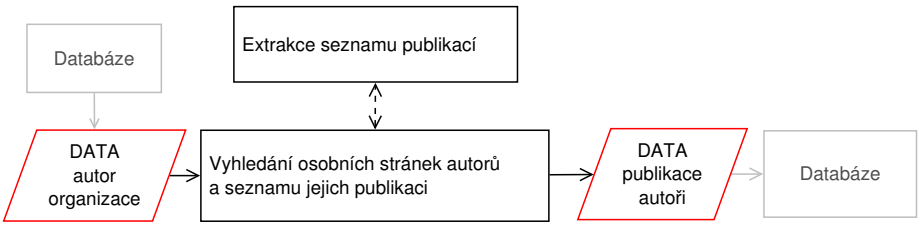

Obrázek 2.5: Modul Extrakce informací o projektech

### 2.3.6 Vyhledávání osobních stránek a publikací autorů

Pokud se v systému objeví nový vědecký pracovník, je často vhodné načíst také jeho osobní údaje a publikace. Modul získá informace za pomoci vyhledávačů Yahoo BOSS[17] , AltaVista[18] , Ask[19] a databází autorů Videolectures[20] , OSTI[21] a Microsoft Academic Search[22]. Z osobní stránky je extrahována adresa elektronické pošty, adresa a seznam publikací, který je dále analyzován za využití podmodulu pro extrakci seznamu publikací [22].

[14] http://liinwww.ira.uka.de/bibliography/
[15] http://cordis.europa.eu/
[16] http://www.epsrc.ac.uk/
[17] http://developer.yahoo.com/search/boss/
[18] http://www.altavista.com/
[19] http://www.ask.com/
[20] http://videolectures.net/
[21] http://www.osti.gov/
[22] http://academic.research.microsoft.com/

Výstupem jsou stránky s osobními údaji, seznamem publikací a údaje o publikacích spolu se jmény autorů. Modul je znázorněn na obrázku 2.6, autorem je Stanislav Heller, podmodul pro extrakci seznamu publikací vytvořila Magdaléna Krygielová. Identifikátorem v řídícím systému je `person_person_data`.

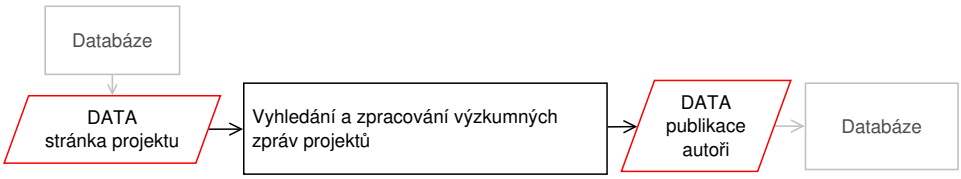



Obrázek 2.6: Modul Vyhledávání osobních stránek a publikací autorů

### 2.3.7 Vyhledání a zpracování výzkumných zpráv projektů

Dalším zdrojem publikací jsou výzkumné zprávy projektů. Na základě informací o projektu modul vyhledá stránku s výzkumnými zprávami, které vzápětí extrahuje [23].

Výstupem modulu jsou publikace a informace o jejich autorech. Modul je znázorněn na obrázku 2.7, autorem je Stanislav Heller a identifikátorem v řídícím systému je `project_url_publication`.

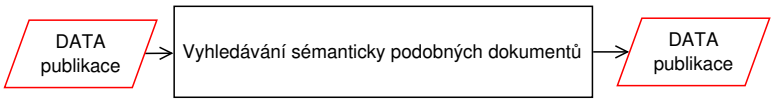



Obrázek 2.7: Modul Vyhledání a zpracování výzkumných zpráv projektů

### 2.3.8 Vyhledání sémanticky podobných dokumentů

Index vytvořený modulem pro indexování dokumentů lze využít pro vyhledání sémanticky podobných dokumentů. Uživateli mohou být poté předloženy dokumenty s podobným tématickým zaměřením.

Vstupem modulu je publikace, na výstupu se objeví dokumenty sémanticky podobné publikaci na vstupu. Modul je znázorněn na obrázku 2.8, autorem je Erik Dresto a identifikátorem v řídícím systému je `publication_similar_publication`.




Obrázek 2.8: Modul Vyhledání sémanticky podobných dokumentů

### 2.3.9 Vyhledání sémanticky podobných termínů

Index je využit také pro vyhledání sémanticky podobných termínů. Podobnosti se určují na základě vyhledávání v matici sémantické blízkosti termínů [24].

Vstupem modulu je termín, výstupem jsou termíny sémanticky nejbližší. Modul je znázorněn na obrázku 2.9, autorem je Ján Novák a identifikátorem v řídícím systému je `keyword_similar_keyword`.

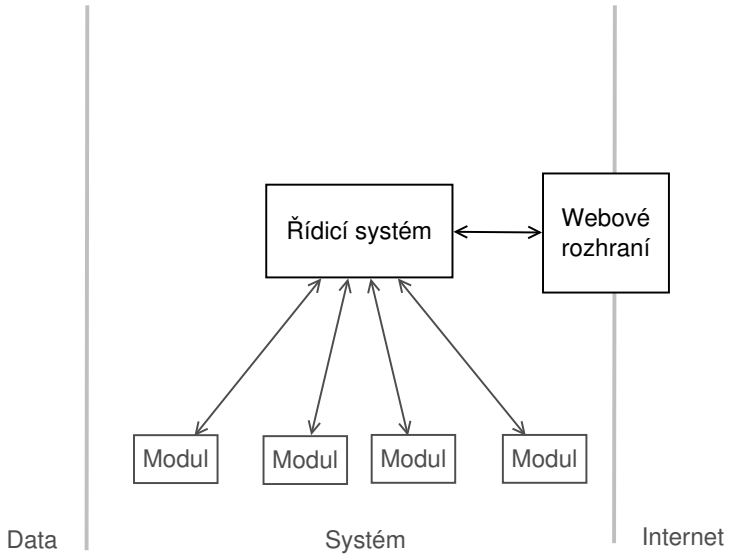

Obrázek 2.9: Modul Vyhledání sémanticky podobných dokumentů

## 2.4 Řídicí systém

Základní součástí projektu ReReSearch je řídicí systém, který zajišťuje spouštění a předávání dat mezi moduly. Základní řídicí systém vytvořil Jan Horák ve své bakalářské práci [17], při následných změnách bude tedy brán jako základ. Kromě zmíněné bakalářské práce jsou další informace k původní verzi řídícího systému k dispozici ve starší dokumentaci [25].

Řídicí systém se skládá ze dvou komponent: démona a webového rozhraní. Démon zajišťuje komunikaci přez XML-RPC s moduly a webové rozhraní představuje prezentační vrstvu, prostředek komunikace s uživatelem.

Blokové schéma systému v původní verzi můžeme graficky vyjádřit obrázkem 2.10.




Obrázek 2.10: Schéma původní verze systému

## 2.5 Cíle práce

Moduly jsou často implementovány jako samostatně fungující systémy, obsahují vlastní databázi a návrh vstupu a výstupu mnohdy v nativních formátech použitých nástrojů. Pro úspěch systému ReReSearch je nutné moduly standardizovat a tak umožnit jejich zapojení do systému. Na základě analýzy stavu řídícího systému lze v souladu se zadáním bakalářské práce stanovit hlavní cíle práce.

### 2.5.1 Centralizace databáze

Nejdůležitějším krokem v integrování modulů do systému ReReSearch je centralizace databáze. Ta by měla sloužit modulům analyzovaným v kapitole 2.3. Při návrhu centralizace databáze lze vycházet ze starší koncepce databáze pro moduly Průvodce konferencí, Extrakce a zpracování informací o projektech a Extrakce a zpracování informací ze seznamu publikací nebo programu konference zpracovaného Peterem Bartošem a Michaelem Angelovem [26]. Autoři také spolupracovali v počáteční fázi nově koncipovaného návrhu. *V této oblasti je nezbytné aktualizovat a dokončit návrh.*

### 2.5.2 Vytvoření archivu souborů

*Zpracovávané soubory jsou uloženy různým způsobem. Je účelné ukládání souborů sjednotit a vytvořit centralizovaný sklad dokumentů, archiv.*

### 2.5.3 Aktualizace souborů v archivu a sledování změn webových stánek

Velká část dat uložených v databázi je extrahována ze stažených souborů a webových stránek. Informace na internetu se však stále mění: autoři přidávají na své stránky nové publikace, vznikají nové organizace a výzkumné skupiny, konají se konference. Budování báze znalostí systémem ReReSearch je dlouhodobého charakteru, uložená data tedy musí být aktualizována a doplněna. *Tato funkce nebyla v řídicím systému průběžně implementována, je tedy nezbytné její vytvoření.*

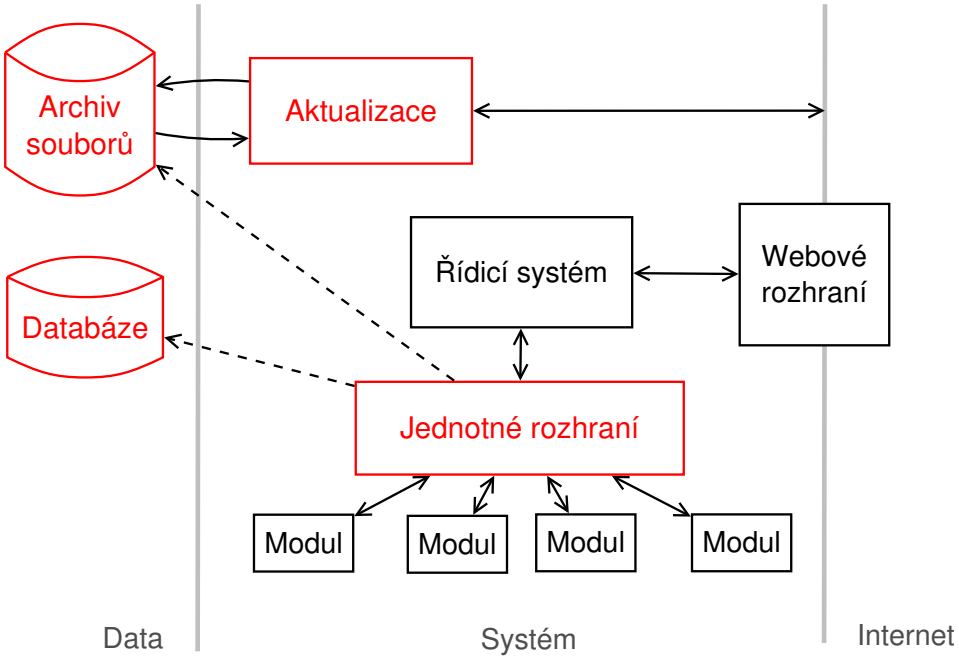
### 2.5.4 Sjednocení rozhraní pro moduly

Za předpokladu, že bude vytvořena centralizovaná databáze, je třeba poskytnout modulům rozhraní pro přístup. Z analýzy vstupů a výstupů modulů v kapitole 2.3 vyplývá, že vstupem modulů je soubor, nebo inicializační data. Na základě vstupů modul exrahuje soubor, zpracuje nebo získá data a ta jsou poté výstupem. *Dalším cílem tedy je vytvořit jednotné rozhraní pro komunikaci modulů.*

### 2.5.5 Vytvoření nástrojů pro práci s databází a archivem souborů

Jednou z výhod sjednocení rozhraní bude možnost vytvoření systémových modulů pracujících s databází a archivem. *Toto bude dalším dílčím cílem.* Modul pro práci s archivem by měl umět ukládat soubory do archivu, tedy vkládat soubory do systému ReReSearch. Modul pro práci s databází by měl být schopný ověřit a zpracovat výstupy modulů navrženým rozhraním a obsažená validní data vkládat do databáze.

Navrhované blokové schéma řídícího systému se nachází na obrázku 2.11, barevně jsou znázorněny nové součásti systému ReReSearch.




Obrázek 2.11: Schéma systému navržené v této práci

# Kapitola 3

# Použité technologie

V kapitole budou vysvětleny nejdůležitější technologie, které budou nezbytné pro dosažení cílů stanovených v analýze (viz. 2.5).

## 3.1 XML

XML (Extensible Markup Language) je jednoduchý, variabilní textový formát sloužící k reprezentaci strukturovaných informací. Vhodnost použití na webu je umožněna odvozením z SGML. Jazyk XML hraje díky své univerzálnosti významnou roli ve výměně dat [27]. Autorem standardu jazyka je World Wide Web Consortium.

### XSD

XSD (XML Schema Definition), nazývaný také XML Schema je jazyk pro definici bloků, ze kterých se může skládat XML dokument. Pomocí XSD lze definovat [28]:

- elementy, které se mohou objevit v dokumentu,
- atributy, které se mohou objevit v dokumentu,
- které elemety jsou potomky,
- pořadí elementů potomků,
- počet elementů potomků,
- jestli musí být element prázdný nebo může obsahovat text,
- datové typy pro elementy a atributy,
- výchozí a pevné hodnoty pro elementy a atributy.

Díky uvedeným vlastnostem lze za pomoci XSD velmi podrobně kontrolovat strukturu XML a může být použito pro odfiltrování XML dokumentů se špatnou syntaxí.

## 3.2 Hashovací funkce

Kryptografická hashovací funkce je deterministická funkce, která dokáže z bloku dat vytvořit řetězec o pevné délce tak, aby platilo, že jakákoliv změna v datech zapříčiní změnu v řetězci. Tomuto řetězci proto říkáme otisk. Jak je zřejmé, hashovací funkce je velmi dobře využitelná ke katalogizaci, protože tvoří ideálně unikátní řetězce.

Reálné otisky však nemusí být pro libovolnou dvojici dat unikátní, protože žádná hashovací funkce není ideální. Chybám v unikátnosti otisků říkáme kolize. Mezi často používané hashovací funkce patří MD5, která má vysokou pravděpodobnost výskytu kolizí. Pro naše potřeby vystačí hashovací funkce SHA-1, kde dle aktuálních výzkumů dochází ke kolizím až po $2^{69}$ operací, jak je uvedeno v [29].

## 3.3 BibTEX

BibTEX[1] je program, který doplňuje typografický systém LaTEX o možnost správy bibliografických referencí. Přináší možnost uložit všechny bibliografické informace do databáze [30], z vlastního textu se pak můžeme jednoduchými příkazy odkazovat na externě uložené reference. Při překladu publikace je automaticky vygenerován seznam citací s odkazy na příslušných místech v textu.

Hlavními výhodami je zrychlení práce s citacemi a automatické formátování bibliografických dat. To umožňuje zaměřit se na vlastní text, odpadá úsilí potřebné na ruční správu referencí a zároveň je jednoduché provádět jakékoliv změny.

Databáze referencí je reprezentována textovým souborem se záznamy předem daných typů o formátu:

```
@BOOK{knuth:tex,
author = "Donald E. Knuth",
title = "The {\TeX}book",
publisher = "Addison-Wesley",
year = "1984",
}
```

Záznamy v databázi jsou tvořeny položkami s určitým typem, v příkladu `BOOK`, který určuje typ citované publikace. Dále se položka skládá z polí, v příkladu `author`, `title`, `publisher` a `year`. Typy polí obsažených v položce jsou definovány typem publikace [31]. Výčet typů publikací a polí v BibTEX záznamech a další informace lze najít v odpovídající literatuře [31, 30].

Reference ve formátu BibTEX, tedy vlastní záznamy publikací, můžeme vidět v digitálních knihovnách, citačních databázích a na osobních stránkách autorů v seznamu publikací. Formát BibTEX je často používaný, protože velmi dobře popisuje typy a informace o publikaci

## 3.4 BibTeXML

BibTEXML[2] je XML (viz. 3.1) reprezentací bibliografických dat ve formátu BibTEX (viz. 3.3). Napravuje skutečnost, že gramatika BibTEX není formálně definována a proto nelze

[1] http://www.bibtex.org/
[2] http://bibtexml.sourceforge.net/

použít pro efektivní manipulaci s daty. Rozšiřuje formát používaný v bibliografické databázi o možnost manipulace a reprezentace dat [32].

## 3.5 Protokol HTTP

Protokol HTTP (Hypertext Transfer Protocol) je protokol aplikační vrstvy pro distribuované, spolupracující hypermediální informační systémy [33]. Komunikace mezi klientem a serverem je realizována systémem zpráv, klient odesílá požadavek (request) a server vrací odpověď (response). Vysvětlíme pojmy hlavička protokolu HTTP a některá pole, která obsahuje. Kompletní popis protokolu je k dispozici v RFC 2616 [33].

### Hlavička

Zprávy obsahují HTTP hlavičku (header), která popisuje data obsažená v těle zprávy. Jednotlivé charakteristiky jsou uloženy ve formě polí (field) některá z nich jsou popsána v tabulce 3.1, komletní seznam lze najít v RFC 2615 [34].

| **Pole** | **Obsah** |
|---|---|
| Content-Lengt | Délka obsahu přenášeného v těle zprávy. |
| Last-Modified | Datum a čas poslední změny dokumentu. |
| ETag | Hodnota charakterizující obsah dokumentu. |
| If-Modified-Since | Pokud nebyl dokument změněn od uvedeného data a času. |
| If-None-Match | Pokud je hodnota charakterizující obsah dokumentu stejná. |

Tabulka 3.1: HTTP Hlavička - pole

### Stavy

V hlavičce každé odpověďi serveru se nachází pole stav (status), které obsahuje stavový kód (status code). Stavový kód vyjadřuje stav odpovědi na požadavek, jsou definovány v RFC 2615 [33] a základní uvedeny v tabulce 3.2.

| **Kód** | **Zpráva** |
|---|---|
| 200 | OK |
| 301 | Moved Permanently |
| 302 | Moved Temporarily |
| 304 | Not Modified |
| 404 | Not Found |

Tabulka 3.2: HTTP Stavy

# Kapitola 4

# Návrh

Podobně jako analýza je při vývoji řídicího systému velmi důležitá formulace podrobného návrhu řešení problému. Je tomu tak z důvodu složitosti implementace některých řešení, zejména rozhraní pro komunikaci modulů. Za použití technologií popsaných v předchozí kapitole bylo navrženo řešení jednotlivých cílů práce (kapitola 2.5).

## 4.1 Databáze

Vzhledem ke komplexnosti podobných systémů této skutečnosti musí odpovídat struktura databáze a použité nástroje.

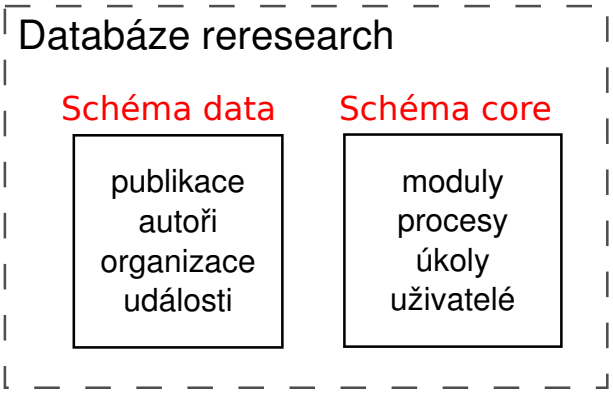
### 4.1.1 Struktura

Mezi nejčastější data ukládaná do databáze patří data získávaná moduly. V databázi budou uložena k dalšímu zpracování nebo prezentaci uživateli. Další databáze v systému je využita samotným řídícím systémem pro uložení úloh a procesů. Tyto části by mělo být možné nezávisle spravovat a optimalizovat, za výhodu považujeme možnost tvořit mezi oběma částmi vazby.

Lze využít jednu z funkcí databázového systému PostgreSQL. Schémata, v některých databázových systémech známa jako katalog, jsou pojmenovanou kolekcí tabulek v rámci jedné databáze. Ve schématech jsou rozlišovány také pohledy, indexy, sekvence, typy dat, operátory a funkce. Vysvětlení uvedených pojmů a podrobnější popis práce se schématy lze nalézt v online dokumentaci PostgreSQL [35] nebo publikaci [36]. V práci budou použita schémata dvě, schéma `data` pro uložení dat, a schéma `core` používané řídícím systémem. Rozdělení databáze na schémata a příklad obsahu, který se v nich může nacházet je vidět na obrázku 4.1

### 4.1.2 Schéma

Zaměříme na schéma `data`. Jak bylo popsáno výše, slouží k uložení dat, se kterými pracují moduly, proto byl jeho obsah navržen tak, aby vyhovoval všem požadavkům. Základními objekty, které je zde možno očekávat, budou publikace, citace, autoři, projekty, organizace, události, lokace a dále pomocné objekty jako klíčová slova a soubory. Jak lze podle počtu modulů a rozsáhlosti systému předpokládat, schéma je komplikované, kompletní verze se nachází v příloze A. V následujícím textu budou postupně komentovány nejdůležitější příklady, u kterých budou uvedeny odpovídající výřezy z plné verze.

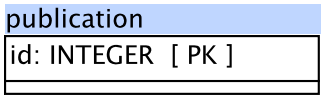



Obrázek 4.1: Rozdělení databáze ReReSearch na schémata

Zavedení jednotného názvosloví v rámci celého schématu usnadní návrh a další rozšiřování databáze, tvorbu rozhraní i modulů s databází pracujících. Pro názvy objektů jsou stanovena následující pravidla: Všechny názvy jsou tvořeny malými písmeny. Použitý databázový systém nerozlišuje velikost písmen u názvů objektů [35]. Z praktických důvodů rovněž nebudou používány mezery, ani jiné nestandardní znaky. Při tvorbě názvů se omezíme na použití číslic, abecedy bez diakritiky a podtržítka _ jako spojovník částí názvů.

### Tabulky

Názvy hlavních tabulek budeme tvořit jednoslovné, jako příklad lze uvést publication, citation, person, event. Každá z nevazebních tabulek obsahuje primární klíč nazvaný `id`.




Obrázek 4.2: Databáze - názvy tabulek

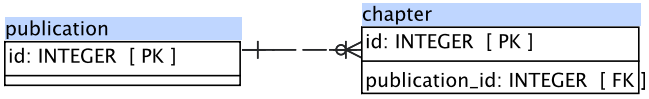
### Vazby a klíče

Pro každou mnohonásobnou vazbu použijeme vazební tabulku pojmenovou ve formátu `j_jmé1_jmé2`, kde `jmé1` a `jmé2` zastupují jména tabulek účastnících se vazby. Pro tvorbu sekce názvu jsou použita maximálně první 4 písmena, pořadí `jmé1` a `jmé2` v názvu je zvoleno podle abecedy. Primární klíč vazební tabulky tvoří cizí klíče tabulek účastnících se vazby.

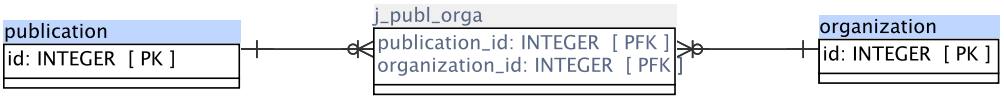
Jednoduché vazby jsou řešeny cizím klíčem v tabulce s vyšší kardinalitou vazby. Pojmenování cizího klíče je definováno jako `jméno1_id`, kde `jméno1` představuje název tabulky obsahující primární klíč. Příklady jsou uvedeny na obrázcích 4.3 a 4.4.

### Role

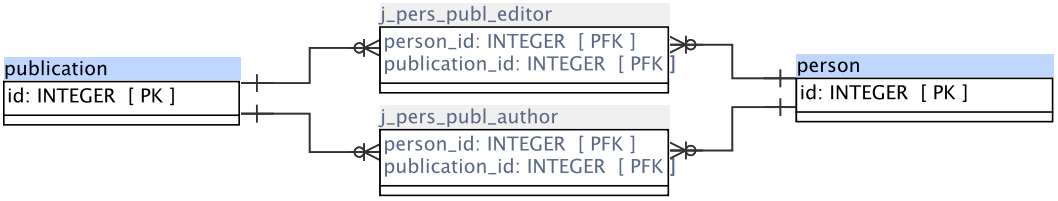
Pokud jsou dvě různé tabulky spojeny navzájem více vazbami, je použita pro rozlišení role vazby. U mnohonásobných vazeb se specifikuje role v názvu vazební tabulky. Místo formátu




Obrázek 4.3: Databáze - jednoduchá vazba




Obrázek 4.4: Databáze - násobná vazba

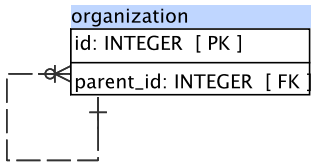



Obrázek 4.5: Databáze - role

`j_jmé1_jmé2` popsaného výše, použijeme `j_jmé1_jmé2_role`, kde `role` zastupuje název role vazby.

### Hierarchie

Při tvorbě databáze může nastat potřeba vyjádřit hierarchii objektů, umístit je do stromové struktury. Problém se řeší použitím cizího klíče stejné tabulky, pojmenovaného jako `parent_id`.

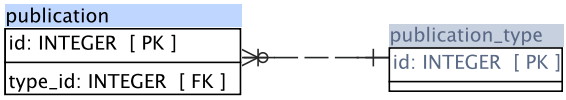



Obrázek 4.6: Databáze - hierarchie

### Typy

Některé z tabulek v databázi charakterizuje typ. V prostředí schématu je tento fakt realizován vytvořením tabulky `jméno_type`, kde `jméno` reprezentuje jméno tabulky, u které chceme určit typ. Tabulka `jméno_type` obsahuje dva sloupce, primární klíč `id` podle pravidel popsaných výše a název typu ve sloupci `type`. Typ objektu nastavíme pomocí cizího klíče `type_id`. Kompletní návrh databáze musí zahrnovat i návrh hodnot typů. Možné typy pro publikace byly vytvořeny na základě BibTEX(viz. 3.3), ostatní kolekce hodnot byly získány analýzou webových stránek. Hodnoty typu události byly získány z plánu, nebo kalendáře událostí nacházejících se na webových stránkách univerzit a vědeckých skupin. Pro

správné použití hodnot typů se jejich definice skládá také z popisu, který je často nezbytný ke správnému rozlišení typů. Typy publikací a událostí se nacházejí v příloze.




Obrázek 4.7: Databáze - typy

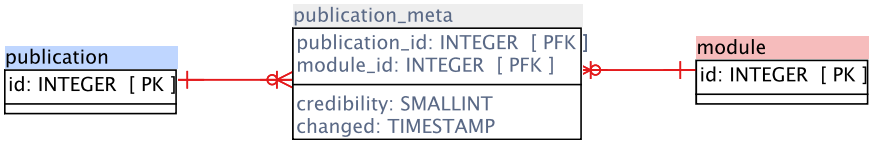
### Značky

Počet záznamů v některých tabulkách bude velmi vysoký, u frekventovaných tabulek lze očekávat miliony položek. Pro usnadnění orientace a jednodnuché kategorizace záznamů byl vytvořen systém značek. Každý z hlavních objektů tvoří vazbu s tabulkou `tag`, která reprezentuje označkování záznamu v tabulce.


Obrázek 4.8: Databáze - značky

### Metainformace

U každého záznamu v databázi musí být dostupné metainformace, zejména o zdrojovém modulu a také čase extrakce. Při zpracování více moduly tak bude dostupná komplentní historie. Uložené metainformace mohou být použity k analýze extrahovaných informací a zlepšení kvality databáze. Podobně jako u značek, každá tabulka tvoří vazbu s tabulkou `jméno_meta`, kde `jméno` představuje jméno tabulky.




Obrázek 4.9: Databáze - metainformace

## 4.2 Archiv

V předchozí kapitole byla popsána databáze sloužící k uložení dat, se kterými moduly pracují. Mnoho modulů však také zpracovává soubory, nejčastěji pak publikace nebo uložené webové stránky. Tuto situaci lze vyřešit vytvořením centralizovaného uložiště pro zmíněné soubory. Jednotlivé moduly tak mají přístup k potřebným souborům, přesto si řídicí systém zachovává nad archivem plnou kontrolu. Začneme specifikací požadovaných parametrů a vlastností archivu.

Hlavním kritériem je kapacita archivu, tedy počet souborů s kterým celý systém pracuje. Na základě velikosti dat dostupných k testování a velikosti databází konkurenčních projektů můžeme tento parametr stanovit. Článků na testování používá systém a moduly přes 1 200 000, analyzované digitální knihovny disponují databází o velikosti nejvýše několik miliónů článků.

Mezi další požadavky na archiv souborů patří:

- Okamžitá dostupnost souborů, rychlý přístup.
- Nenáročná implementace.
- Uchování původních jmen souborů z důvodu možného exportu souborů uživateli.
- Nepřetížení souborového systému.
- Možnost označení souborů značkami.

### 4.2.1 Úložný prostor a souborový systém

Prvním velkým omezením při realizaci archivu jsou dostupné systémové prostředky, zejména kapacita úložného prostoru a použitý souborový systém. Publikace v elektronické formě se nejčastěji nachází ve formátu pdf. Na základě dat dostupných k testování můžeme zjistit, že průměrná velikost článku ve formátu pdf činí 550 KB, uložení 2 000 000 článků zabere 1,1TB diskového prostoru.

Dalším úskalím je problém vysokého počtu souborů v jednom adresáři, který je závislý na použitém souborovém systému. Pro vývoj ReReSearch se využívá souborový systém XFS, který ukládá soubory v adresáři pomocí B+ tree, má tedy podporu pro tzv. Large Directories (složky které mohou obsahovat velký počet souborů bez ztráty na výkonnosti).

### 4.2.2 Komprimované soubory

Při práci s publikacemi je možno se setkat také s formáty obsahujícími komprimovaná data, mezi testovacími články jich můžeme najít asi 10% [37]. Nejrozšířenější z formátů představuje postscript soubor uložený kompresí gzip, jedná se o soubory s příponou .ps.gz, při použití této metody každý soubor obsahuje právě jednu publikaci. Dalším formátem jsou archivy využívané k poskytování většího množství souborů v jednom balíčku, často se jedná o soubor článků se společnou tématikou. S komprimovanými soubory se pracuje obtížněji, proto je nezbytné soubory při vstupu do systému automaticky extrahovat. Z původních souborů musí být uchován pouze název, protože při exportu může být vyžadována opětovná komprese.

### 4.2.3 Vstup souborů do systému

Soubory v systému ReReSearch lze rozdělit do dvou typů. Do první kategorie spadají soubory, které si přeje uživatel do systému importovat. V tomto případě se může jednat například o přenosné médium obsahující články z konference, nebo adresář s oblíbenými články. Uživatel by také měl mít možnost nahrávat jednotlivé soubory. Druhou kategorií publikací jsou soubory, které systém ReReSearch získá a automaticky zařadí do databáze. Tato situace může nastat, pokud je soubor získán modulem a je uveden v jeho výstupu.

Pro vložení souborů do systému bude nutné vytvořit modul, který bude zajišťovat dekompresi souborů a umístění do archivu. Vstupem modulu jsou jména souborů, složka, nebo webová adresa. Modul zpracuje uvedené soubory, zařídí jejich integraci do systému a na výstupu vrátí seznam vložených souborů, které následně mohou být zpracovány moduly.

## 4.3 Aktualizace

Systém pracuje s velkým množstvím webových stránek a souborů, vzhledem k vyšší časové náročnosti extrakce informací je tedy realizace opětovným zpracováním všech zdrojů technicky neproveditelná. Jako jediné východisko se jeví zpracovávat pouze zdroje, které byly změněny. Úkolem aktualizačního systému je pravidelně kontrolovat zdroje dat a v případě potřeby inicializovat jejich znovuzpracování moduly pro získávání informací.

Webových stránek a souborů se v systému nachází velké množstí, proces aktualizace tak musí být co nejrychlejší. Nalezení konkrétních změn v obsahu je časově náročné, proto lze zvýšit efektivitu omezením analýzy na stránky, které byly označeny za změněné.

Aktualizační systém provádí dvojí klasifikaci, postup při aktualizaci stránky je následující:

1. Načtení informací o stránce z databáze.
2. Rozhodnutí o tom, zda byla stránka od poslední návštěvy změněna.
3. Nalezení konkrétních změn v obsahu.

### 4.3.1 Rozhodnutí o změně

Do první skupiny metod použitých pro rozhodnutí o změně stránky patří postupy, které pracují přímo s protokolem HTTP (viz. 3.5), nepotřebují tedy vlastní obsah stránky, pouze hlavičku. Na efektivitu stahování stránek má hlavní vliv rychlost, která je v některých případech mnohonásobně pomalejší než dovoluje kapacita linky a proto je nutné stahování realizovat ve více paralelně bežících vláknech. Při použití následujících metod se přenáší pouze velmi malé množství dat a to je činí velmi efektivní.

#### If-None-Match

Při odeslání požadavku na vrácení obsahu stránky je do hlavičky přidáno pole `If-None-Match` s hodnotou `ETag` poslední verze stránky v archivu. Pokud se tato hodnota shoduje s hodnotou na serveru, je vrácen pouze návratový kód 304 bez obsahu.

#### If-Modified-Since

Metoda velmi podobná jako předchozí. Do hlavičky požadavku je nastavena hodnota pole `If-Modified-Since` na čas, ve kterém byla stránka naposledy kontrolována. Pokud stránka nebyla od této doby změněna, vrací se kód 304 bez obsahu. Tato metoda velmi dobře funguje u statických stránek.

#### Content-Length

Z hlavičky odpovědi je přečteno pole `Content-Length`. Pokud se shoduje s hodnotou uloženou při poslední návštěvě stránky, je velká pravděpodobnost, že obsah stránky není nutné znovu stahovat, rozhodnutí záleží na konkrétním nastavení parametrů.

Pokud o změně nerozhodne žádná z uvedených metod, je stažen obsah stránky. Dále jsou použity následující postupy, v případě že o změně stránky stále není rozhodnuto, je považována za změněnou.

#### Otisk obsahu

U každé aktualizované stránky je uložen otisk poslední kontrolované verze stránky. Je vypočítán otisk i pro aktuální verzi a oba porovnáme. Pokud jsou stejné, tak stránka změněna nebyla.

#### Délka obsahu

Obdobná metoda jako Content-Length z předchozí skupiny, zde však musíme získat délku z obsahu stažené stránky.

#### Počet klíčových značek

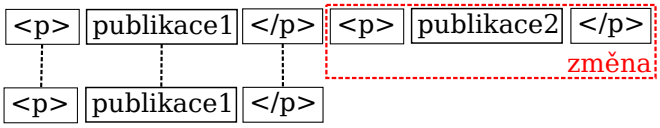
Načteme obsah aktuální stránky a také poslední stažené verze, u obou porovnáme počet značek (tagů), které mohou obsahovat informace - například počet odkazů, počet nadpisů.

### 4.3.2 Nalezení změn

U změněných stránek přichází na řadu určení konkrétních změn. Nejdříve jsou z obsahu stránky odstraněny nepodstatné informace – komentáře, hlavička, vložené skripty a kaskádové styly.

Dále je dokument rozdělen do úseků podle následujícího pravidla: v každém úseku se nachází pouze informace nacházející se na stránce, nebo pouze jedna formátovací značka (tag). Uvedeným postupem je odděleno formátování dokumentu od vlastních dat, to prospívá k lepším výsledkům rozpoznání změn a jejich jednoduchému zpracování. Předchozí postup je použit i u posledního dostupného obsahu stránky. Úseky stránky z archivu a aktuálního obsahu jsou porovnány algoritmem, který zajistí provázání vzájemně si odpovídajících úseků. Získáme tak pouze úseky, které byly od minulé návštěvy změněny, podobně jako na obrázku 4.10.

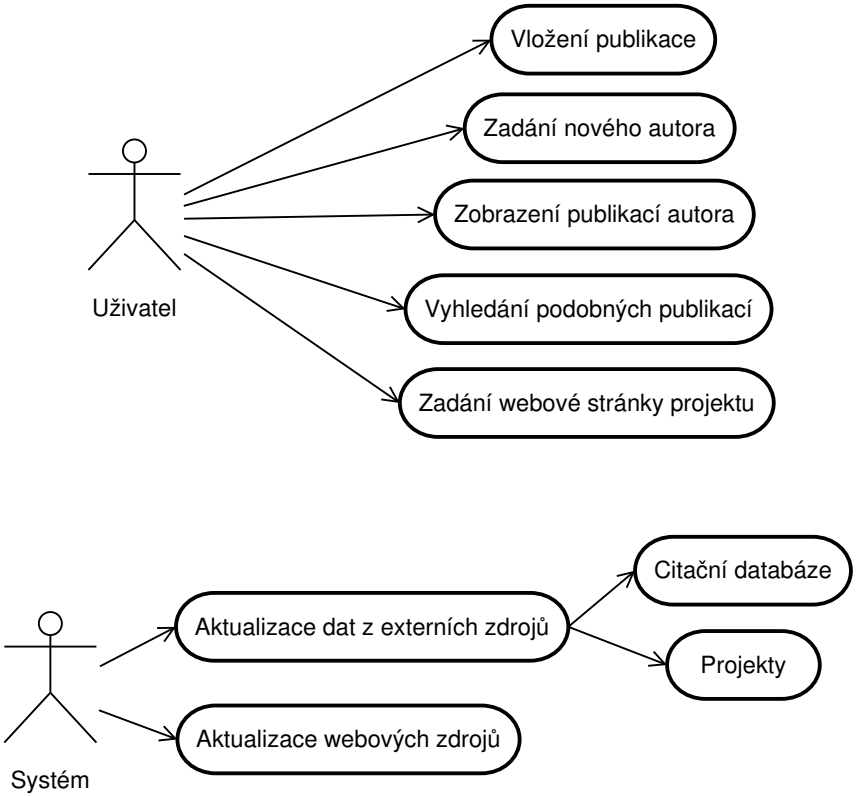
Popsaným postupem jsou rozpoznány konkrétní změny, které jsou případně dále filtrovány podle nastavených parametrů. Správným nastavením filtrace změn lze reagovat na pravidelné změny v obsahu stránky, mezi které patří například obrázky s reklamou, vestavěné hodiny na stránce a počítadlo návštěv. Lze také nastavit minimální počet znaků, které musí být změnněny aby byla stránka klasifikována jako změněná.




Obrázek 4.10: Nalezení změn - porovnání úseků

## 4.4 Moduly

V kapitole 2.3 byly popsány moduly dostupné v systému ReReSearch. Pro správný návrh jednotného rozhraní a demonstračních dat je vhodné uvést případy užití systému, kde budou stanoveny účastnící se moduly. Diagram případů užití se nachází na obrázku 4.11. Níže jsou popsány vybrané specifikace případů užití.




Obrázek 4.11: Diagram případu užití

| **Případ užití: Aktualizace dat z externích zdrojů** |
| --- |
| **Účastníci:**<br>Systém |
| **Vstupní podmínky:**<br>1. Je naplánovaná aktualizace databáze? |
| **Tok událostí:**<br>1. Systém spouští modul *Začlenění dat citačních databází.*<br>2. Systém spouští modul *Extrakce a zpracování informací o projektech.* |
| **Následné podmínky:**<br>1. Moduly zajistí aktualizaci databáze. |

| **Případ užití: Zobrazení publikací autora** |
| --- |
| **Účastníci:**<br>Uživatel |
| **Vstupní podmínky:**<br>1. Uživatel vybral autora. |
| **Tok událostí:**<br>1. Modulem *Vyhledání osobních stránek autorů a seznamu jejich publikací* je získána domovská stránka uživatele a seznam jeho publikací.<br>2. Získané publikace jsou zobrazeny uživateli.<br>3. Získané publikace jsou uloženy do databáze.<br>4. Dostupné elektronické verze publikací jsou uloženy do archivu souborů.<br>5. Elektronické verze publikací jsou zpracovány modulem *Převod publikací do textového tvaru a extrakce metainformací.*<br>6. Metainformace jsou uloženy do databáze.<br>7. Z textové podoby publikace jsou modulem *Automatické navrhování klíčových slov* extrahována klíčová slova.<br>8. Klíčová slova jsou použita pro vytvoření indexu modulem *Indexace textových dokumentů.* |
| **Následné podmínky:**<br>1. Uživateli jsou zobrazeny publikace od vybraného autora.<br>2. Získané publikace a metainformace jsou přidány do databáze.<br>3. Získané soubory jsou uloženy do archivu souborů.<br>4. Textová podoba publikace je indexována. |

| **Případ užití: Vložení publikace** |
| --- |
| **Účastníci:**<br>Uživatel |
| **Vstupní podmínky:**<br>1. Uživatel nahrál do systému elektronickou publikaci. |
| **Tok událostí:**<br>1. Publikace je uložena do archivu souborů.<br>2. Publikace je zpracována modulem *Převod publikací do textového tvaru a extrakce metainformací.*<br>3. Získané metainformace jsou předloženy uživateli.<br>4. Z textové podoby publikace jsou modulem *Automatické navrhování klíčových slov* extrahována klíčová slova.<br>5. Klíčová slova jsou použita pro vytvoření indexu modulem *Indexace textových dokumentů.* |
| **Následné podmínky:**<br>1. Publikace je uložena v archivu souborů.<br>2. Uživateli jsou zobrazeny informace o nahrané publikaci.<br>3. Získaná data jsou zařazena do databáze.<br>4. Textová podoba publikace je indexována. |

| **Případ užití: Zadání webové stránky projektu** |
| --- |
| **Účastníci:**<br>Uživatel |
| **Vstupní podmínky:**<br>1. Uživatel zadal do systému webovou stránku projektu. |
| **Tok událostí:**<br>1. Modul *Vyhledání a zpracování výzkumných zpráv projektu* vyhledá publikace související s projektem.<br>2. Extrahované publikace jsou zobrazeny uživateli.<br>3. Extrahované publikace jsou uloženy do databáze.<br>4. Dostupné elektronické verze publikací jsou staženy do archivu souborů.<br>5. Elektronická verze publikace je převedena na text a zpracována modulem *Převod publikací do textového tvaru a extrakce metainformací.*<br>6. Extrahované metainformace jsou uloženy do databáze a textová podoba publikace do archivu souborů.<br>7. Z textové podoby jsou modulem *Automatické navrhování klíčových slov* extrahována klíčová slova.<br>8. Klíčová slova jsou použita pro vytvoření indexu modulem *Indexace textových dokumentů.* |
| **Následné podmínky:**<br>1. Uživateli jsou zobrazeny výzkumné zprávy projektu.<br>2. Extrahovaná data jsou uložena do databáze.<br>3. Získané soubory jsou uloženy do archivu souborů.<br>4. Textová podoba publikace je indexována. |

# Kapitola 5

# Implementace

## 5.1 Jazyk

Systémy na vkládání dat do databáze, a aktualizaci byly implementovány v jazyce python. Výběr jazyka byl určen původní verzí řídicího systému, volba se ukázala vhodná pro daný učel. Jazyk python poskytuje mnoho modulů pro práci s XML-RPC, XML, XSD a práci s PostgreSQL databází. Standardní instalace však některé z použitých modulů neobsahuje, proto je nutné chybějící moduly do systému doinstalovat.

## 5.2 Aktualizace

Archiv posledních verzí stránek je realizován adresářovou strukturou v souborovém systému založenou na podobném principu jako archiv souborů. Poslední verze webových stránek spolu s metadaty jsou uloženy v podobě serializovaných objektů, výhodou je tedy velmi snadné použití a vysoká efektivita. Objekty byly serializovány za pomocí modulu cPickle[1], mezi další použité moduly patří `threading`, `hashlib`, `urllib2`. Pro rozpoznání změn v obsahu stránek rozdělených na úseky (viz. 4.3.2) je použit algoritmus `SequenceMatcher`.

## 5.3 Struktura složek

Domovský adresář systému je `/mnt/data2/reresearch` na serveru `athena3.fit.vutbr.cz`. V tabulce 5.1 jsou vysvětleny funkce jednotlivých podadresářů. Adresářová struktura modulů obsahuje moduly dle identifikátorů zmíněných v analýze 2.3, adresář s modulem má strukturu popsanou v tabulce 5.2.

## 5.4 XML Rozhraní

V kapitole 4.1 byla na základě potřeb modulů navržena centralizovaná databáze. Zbývá specifikovat rozhraní, kterým by mohly moduly databázi využívat. Při analýze modulů byla věnována velká pozornost vstupům a výstupům, protože slouží jako podklad pro návrh rozhraní, kterým budou moduly komunikovat a předávat data. Při návrhu rozhraní si klademe za cíl co největší srozumitelnost a jednoduchost, aby bylo snadné implementovat vstupy a výstupy. Přesto budeme vyžadovat, aby bylo možné reprezentovat libovolně provázaná data

[1] http://docs.python.org/release/2.5/lib/module-cPickle.html

| | |
|---|---|
| `core.daemon` | Daemon řídícího systému. |
| `core.www` | Webové rozhraní řídícího systému. |
| `db` | Databáze: skripty a nástroje pro spávu databáze. |
| `dev` | Závislosti a externí programy. |
| `export` | Výstupní složka pro dokumenty. |
| `ext` | Společné součástí pro systémové moduly. |
| `files` | Uložiště souborů. |
| `import` | Vstupní složka pro dokumenty. |
| `log` | Složka obsahující log soubory. |
| `modules` | Adresářová struktura modulů. |
| `public.www` | Webový server s nástroji. |
| `tmp` | Adresář s dočasnými soubory řídícího systému. |
| `tools` | Pomocné skripty a nástroje. |
| `reresearch` | Skript pro spuštění a zastavení systému. |

Tabulka 5.1: Adresářová struktura systému ReReSearch.

| | |
|---|---|
| `bin` | Spustitelný soubor modulu. |
| `data` | Doplňová data modulu (Skripty, schémata, dokumentace). |
| `input.test` | Testovací vstupy modulu. |
| `input` | Vstupy modulu od řídícího systému. |
| `output.test` | Testovací výstupy modulu. |
| `output` | Výstupy modulu řídícímu systému. |
| `tmp` | Adresář pro dočasné soubory. |

Tabulka 5.2: Adresářová struktura modulu.

uložitelná do databáze a byla podporována práce se soubory v archivu. Realizaci rozhraní bude popsána na základě elementárních problémů, které umí řešit a bude využita struktura a pojmenování databáze. Autorům jednotlivých modulů jsou pro návod k implementaci předloženy demonstrařní vstupy a výstupy pokrývající všechny možné kombinace dat.

### Sloupec

Sloupce v tabulce budou reprezentovány podelementy, vlastní hodnota je uložena v atributu value.
**Situace**: Publikace má název Publication Title.

```
<publication>
  <title value="Publication Title" />
</publication>
```

### Vztahy

Vztahy mezi tabulkami v databázi reprezentuje zanoření elementů XML. Při analýze XML rozhraní je k dispozici kompletní struktura databáze, snadno tedy lze z názvů elementů

určit o kterou vazbu se jedná. Díky vhodnému návrhu pojmenování databázových objektů bude snadno rozpoznatelná také kardinalita vazeb a použity odpovídající vazební tabulky a cizí klíče. Vztahy jsou od atributů rozlišeny přítomností atributu value.
**Situace**: Publikace má název a je v ní citována publikace s názvem Cited Title.

```
<publication>
  <title value="..." />
  <citation>
    <title value="Cited Title" />
  </citation>
</publication>
```

### Role

V případě, že je mezi dvěma tabulkami více vztahů, je nutné upřesnit vztah použitím role. Obdobným způsobem je problém řešen v XML: vazba je tvořena podle pravidel uvedených v předchozím bodě, je však nutné přidat vazební entitě atribut role s hodnotou jméno role.
**Situace**: Publikace má autora/osobu s křestním jménem John.

```
<publication>
  <title value="..." />
  <person role="author">
    <first_name value="John" />
  </person>
</publication>
```

### Existující data

Prostřednictvím XML musí být možné odlišit data která jsou již v databázi od dat nových. Na výstupu se mohou data již existující v databázi objevit pokud tvoří vazbu, nebo pokud je chceme měnit. Primární klíče nelze reprezentovat jako normální sloupce.
**Situace**: Publikace je uložena v databázi pod identifikátorem 316.

```
<publication id="316">
  <title value="..." />
</publication>
```

### Zdrojový modul

Na základě návrhu z kapitoly 4.1.2 je u nově získaných dat nutné na výstupu specifikovat zdrojový modul pomocí atributu `source-module`, hodnotou je identifikátor modulu v řídicím systému. Specifikace zdrojového modulu je dědičná, je možné ji uvést pouze u elementu na nejvyšší úrovni.
**Situace**: Publikace s názvem New je extrahována modulem `publication_file_meta`.

```
<publication source-module="publication_file_data">
  <title value="New" />
</publication>
```

### Důvěryhodnost

U nově získaných dat je též nutné uvést důvěryhodnost. Hodnota důvěryhodnosti je uvedena v atributu credibility jako číslo od 0 do 1, vyjadřující pravděpodobnost, s kterou jsou data správná.
**Situace**: Publikace s názvem New, získaná modulem `publication_file_meta`, je s 75% pravděpodobností správná.

```
<publication source-module="publication_file_data" credibility="0.75">
  <title value="New" />
</publication>
```

### Značka

Podle návrhu z kapitoly 4.1.2 je možné data označit značkou. Prostřednictvím XML je tato skutečnost vyjádřena atributem tag u odpovídajícího elementu.
**Situace**: Publikaci s názvem Tag je přiřazena značka csx.

```
<publication tag="csx">
  <title value="Tag" />
</publication>
```

### Více hodnot v atributu

V některých případech je nutné u atributů specifikovat více hodnot, které lze oddělit čárkou.
**Situace**: Publikaci s názvem Tag jsou přiřazeny značky csx a twocolumn.

```
<publication tag="csx, twocolumn" />
  <title value="Tag" />
</publication>
```

### Typy dat

V návrhu databáze byla popsána také specifikace typů (viz. 4.1.2), pro podporované tabulky je tedy možné určit typ za pomocí atributu type. Hodnotou je odpovídající typ podle tabulek v příloze C.
**Situace**: Publikace s názvem Type je typu article.

```
<publication type="article">
  <title value="Type" />
</publication>
```

## 5.5 Uložení dat reprezentovaných v XML do databáze.

Před analýzou XML souboru ke zpracování musí být nejdříve načtena struktura databáze. Jako zdroj může sloužit vlastní databázový server - v této implementaci PostgreSQL[2], nebo zdrojový soubor z návrháře schéma databáze Power*Architect[3]. Přidání dalších zdrojů

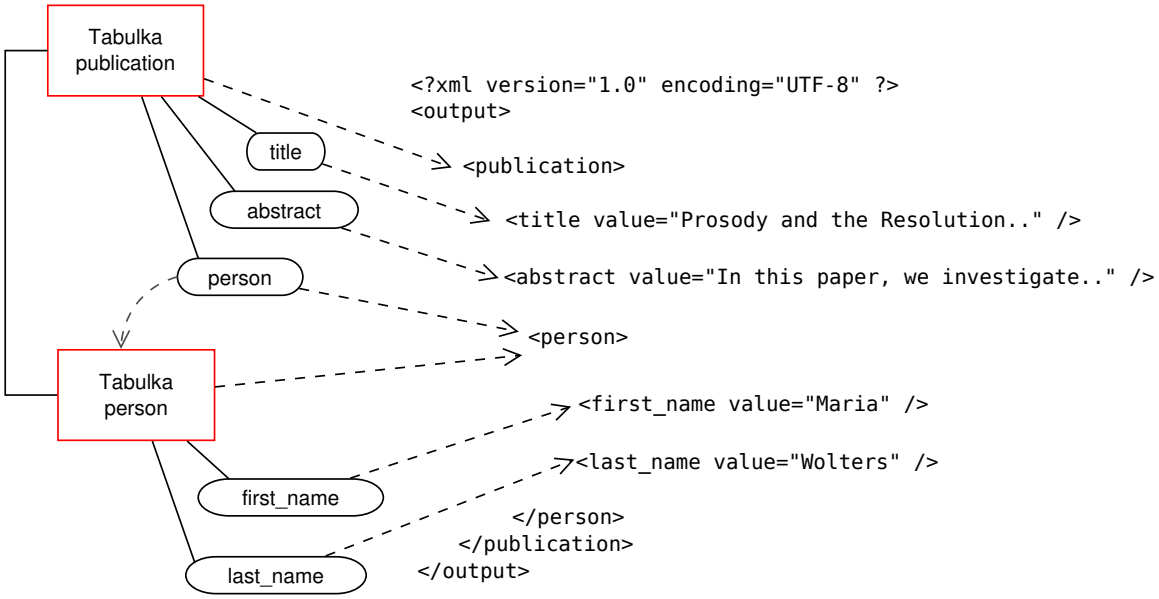
[2] http://www.postgresql.org/
[3] http://www.sqlpower.ca/page/architect

schéma je velmi jednoduché. Schéma je reprezentováno jako seznam objektů tabulek, se stejnou logikou jako v databázi. Pro každou tabulku a sloupec jsou načteny také odpovídající vlastnosti jako klíče, vazby, omezení, datové typy, možnost zadat NULL hodnoty.

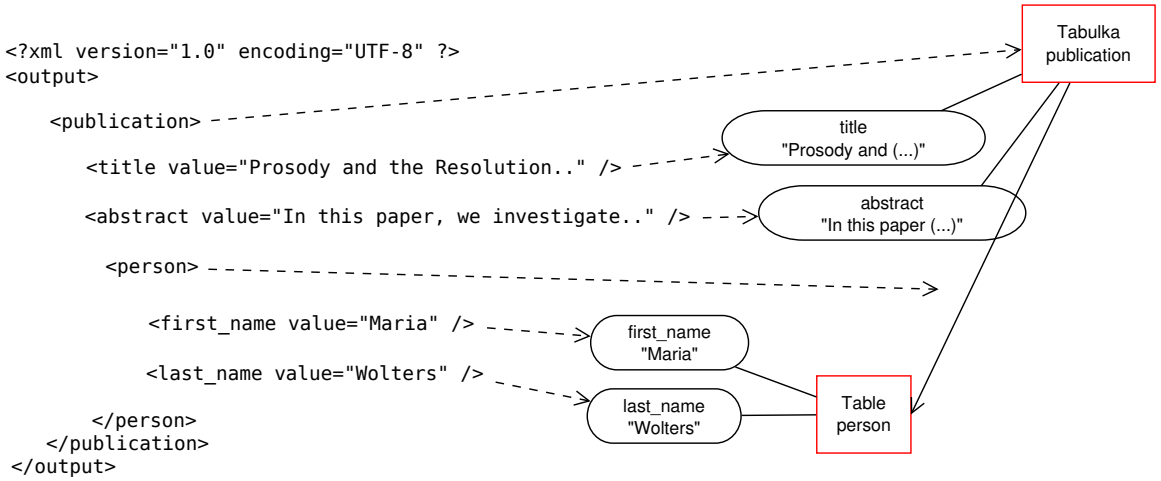
Po načtení struktury databáze následuje fáze čtení XML souboru. Ten je načten do objektového modelu dokumentu (DOM) a postupně procházen. Celý algoritmus bude popsán na základě příkladu XML k analýze. Na obrázku 5.1 je znázorněno načtené schéma databáze, na obrázku 5.2 poté datový strom vznikající z XML.

```
<?xml version="1.0" encoding="UTF-8" ?>
<output>
   <publication>
      <title value="Prosody and the Resolution.." />
      <abstract value="In this paper, we investigate.." />
      <person>
         <first_name value="Maria" />
         <last_name value="Wolters" />
      </person>
   </publication>
</output>
```

1. Na nejvyšší úrovni, jako kořenový element, je očekáván element `output`.
2. Jako podelement `output` očekáváme tabulky. Načteme tedy element `publication` a vyhledáme jej ve schématu mezi tabulkami.
3. Ve výsledném datovém stromu vytvoříme tabulku `publication`. Dále postupně načítáme subelementy publication, ve kterých očekáváme sloupce nebo vazby tabulky.
4. Element `title` identifikujeme jako sloupec, z atributu `value` poznáme hodnotu a tu nastavíme u odpovídajícího sloupce v datovém stromu. Obdobně postupujeme pro element `abstract`.
5. Dalším elementem je `person`, který se nenachází mezi sloupci tabulky `publication`. Podle informací o tabulce `publication` získaných ze schématu představuje vazbu s tabulkou `person`. Do datového stromu přidáme tabulku person podřízenou publication - odtud datový strom.
6. Pro podelement `person` postupujeme obdobným způsobem, naplníme sloupce hodnotami.
7. Naplněný datový strom můžeme sestupně procházet a vkládat záznamy do tabulek. Vazby zajistíme propojením pomocí klíčů, případně vazebními tabulkami. Díky hierarchické struktuře však budeme mít vždy k dispozici klíč potřebný pro vazbu.




Obrázek 5.1: Algoritmus analýzy XML - schéma databáze




Obrázek 5.2: Algoritmus analýzy XML - datový strom

## 5.6 Generování XSD

Pro ověření správnosti komunikačních XML souborů vytvářených moduly lze vhodně použít XML schéma. Návaznost XML rozhraní umožňuje XSD vygenerovat na základě schéma databáze. Je použit následující algoritmus:

1. Načtení struktury databáze.
2. Vytvoření elementu na nejvyšší úrovni s názvem `input` nebo `output` s výčtem jeho možných podelementů, který tvoří všechny nevazební tabulky.
3. Vytvoření podelementů/tabulek, následujícím postupem:
   (a) Vygenerování možných atributů elementu: `role`, `id`, `source-module`, `credibility` a `tag`.
   (b) Vytvoření možných podelemetů představující sloupce.
   (c) Přidání odkazů na jiné elementy reprezentující vazby.
   (d) Přidání odkazu na typ, pokud je u konkrétní tabulky k dispozici. Vytvoření nového typu pro odpovídající tabulku.

# Kapitola 6

# Testování

V této kapitole se nachází zhodnocení implementace a jsou stanoveny orientační časové nároky pro vybrané akce systému ReReSearch.

## 6.1 Archiv souborů

Pro testování archivu souborů byly k dispozici dvě testovací sady publikací. Počty článků a průměrné velikosti jsou zobrazeny v tabulce 6.1.

| Zdroj | Počet | Celková velikost | Průměrná velikost |
|---|---|---|---|
| Sbírka testovacích článků | 160 500 | 46,9 GB | 307 KB |
| Články na CiteSeer$^X$ [37] | 859 529 | 501 GB | 611 KB |

Tabulka 6.1: Archiv - velikost souboru

Formáty souborů zastoupené v sadě testovacích článků se nachází v tabulce 6.2.

| Formát | Počet | Část z celku |
|---|---|---|
| pdf | 688019 | 80,0% |
| ps | 88370 | 10,4% |
| ps.gz | 70985 | 8,2% |
| ps.z | 12155 | 1,4% |
| **celkem** | 859529 | 100% |

Tabulka 6.2: Archiv - významné formáty souborů

## 6.2 Aktualizace

Systém aktualizace byl testován na seznamu 60 000 stránek zachycených se služby Delicious[1], komunitní služby pro publikaci odkazů. Stránky v uvedeném seznamu mají často složitější formátování a větší počet reklam, než cílové stránky s publikacemi, nebo osobní stránky autorů, představují tedy složitější testovací sadu. Systém aktualizace umí pracovat

[1] http://delicious.com/

také v módu ověření dostupnosti webové stránky, který může být použit pro vyřazení neplatných odkazů. Výsledky testů jsou znázorněny v tabulce 6.3. Sledování 60 000 webových stránek vyžaduje 5 GB diskového prostoru pro uložení archivu, průměrná velikost stránky pak je 85 KB.

| **Popis testu** | **Počet stránek** | **Čas** | **Počet vláken** | **Rychlost** |
|---|---|---|---|---|
| ověření dostupnosti | 61 200 | 32 min | 30 | 30 stránek/s |
| hledání rozdílů | 61 200 | 95 min | 30 | 10 stránek/s |

Tabulka 6.3: Výsledky testování systému aktualizace

## 6.3 Rozhraní

Vstupy a výstupy modulů pracujících v systému ReReSearch by měly být navrženy dle pravidel tvorby XML rozhraní. Pro každý z modulů byly vytvořeny demonstrační vstupy a výstupy. Pro testování výstupů modulů se nabízí možnost validace výstupu za pomoci XSD schéma, v případě že výstup bude ověřen můžeme jej zpracovat modulem pro vkládání dat do databáze.

## 6.4 Případy užití

Vzhledem k tomu, že se u modulů nepodařilo implementovat rozhraní, případy užití byly testovány v simulovaném prostředí. Nyní budeme testovat, zda navržené případy užití vedou k požadovanému cíli. Budou popsány akce, které jsou provedeny a vybrána demonstrační data, která jsou uložena na médiu v příloze.

### Zadání jména autora

1. Vyhledávání osobních stránek a publikací autorů

   **Vstup**: jméno autorky Ella Bingham

   **Výstup**: stránka s publikacemi, 27 extrahovaných publikací

   **Čas**: 10 s

2. Stažení plné verze dostupných 22 publikací

   **Čas**: 8 s

3. Převod publikací do textového tvaru a extrakce metainformací

   **Vstup**: 22 stažených publikací

   **Výstup**: 22 publikací v textové podobě, metainformace z 22 publikací

   **Čas**: 11 s převod na text, 55 s extrakce metainformací

4. Identifikace klíčových slov

   **Vstup**: 22 stažených publikací

   **Výstup**: 22 publikací v textové podobě, metainformace z 22 publikací

**Čas**: 21 s

5. Indexace textových souborů

   **Vstup**: klíčová slova

## Orientační časové nároky

Na základě informací o modulech a návrhu případů užití lze stanovit orientační časové nároky pro akce systému.

| **Akce** | **Čas** | **Počet/1h** |
|---|---|---|
| Převod do textového tvaru | 0,3 s | 12000 |
| Extrakce metainformací z publikací | 1,7 s | 2093 |
| Převod publikací do textového tvaru a extrakce metainformací | 2,0 s | 1800 |
| Vyhledávání a zpracování výzkumných zpráv projektů | 28,0 s | 128 |
| Vyhledání osobních stránek autorů | 11,5 s | 313 |
| Vložení souborů do systému | 0,013 s | 275142 |
| Případ užití *Zobrazení publikací autora* (viz. 4.4) | 105,0 s | 34 |

Tabulka 6.4: Orientační časové nároky

# Kapitola 7

# Závěr

Cílem práce bylo vytvořit centralizovanou databázi, archiv souborů, aktualizační službu a jednotné rozhraní pro moduly systému ReReSearch. Uvedené cíle měly umožnit integraci modulů do systému ReReSearch a připravit celý systém pro praktické použití.

Díky množství rozdílných modulů pro získávání informací bylo v průběhu návrhu databáze nutné provádět neustálé aktualizace. To zpomalovalo práci na ostatních součástech systému, které s databází pracují. Přínosem dokončení návrhu databáze je pokrok v integraci modulů do systému ReReSearch. Návrh pojmenování objektů v databázi podle předem stanovených pravidel významně ulehčil následující specifikaci rozhraní pro komunikaci modulů a rovněž zjednodušil návrh a implementaci modulu komunikujícího s databází.

Vytvoření archivu souborů umožňuje snadnou správu souborů v systému, zároveň však zachovává jednoduchý způsob přístupu k souborům. Byl otestován za použití rozsáhlé sady publikací, je tedy připraven pro praktické použití.

Systém aktualizace byl navržen, implementován a testován bez centralizované databáze a archivu souborů, před praktickým použitím je nezbytné jej modifikovat pro potřeby systému ReReSearch. Přínosem aktualizačního systému pro projekt ReReSearch je zvýšení efektivity opakované extrakce dat. Díky univerzálnímu návrhu je však systém možný použít také samostatně pro sledování změn webových stránek, efektivnímu stahování souborů nebo kontrole dostupnosti webových stránek.

Jednotné rozhraní pro výměnu dat mezi moduly bylo již implementováno některými moduly a otestováno. Proces zavádění je však díky většímu počtu autorů modulů složitější, lze jej však ulehčit důkladnou dokumentací a připravením demonstračních vstupů a výstupů pro moduly. Pro projekt ReReSearch znamená zavedení jednotného rozhraní pro výměnu dat sjednocení modulů a je významným krokem k jeho spuštění.

Při dalším vývoji součástí projektu ReReSearch je vhodné využít databáze, archivu souborů a rozhraní navržených v této práci. V případě nutnosti změn v návrhu databáze jsou připravena pravidla pojmenování objektů. Jako nejdůležitější krok se jeví implementace rozhraní moduly, po které mohou být důkladně otestovány a prakticky používány. Ke kvalitě dat a pohodlné práce se systém by mohlo přispět zavedení ontologií do návrhu databáze, zejména pak geografické ontologie. Před spuštěním projektu by bylo také vhodné vytvořit veřejnou stránku s informacemi o projektu. Při práci na řídicím systému projektu této velikosti obtížné analyzovat ostatní komponenty, proto je nezbytné všechny postupy důkladně dokumentovat.

Vzhledem k zařazení této bakalářské práce do projektu je pravděpodobné, že bude použita v praxi a dále rozšířena. Ostatní součásti projektu ReReSearch jsou také vytvářeny

převážně v rámci bakalářských a diplomových prací. Paralelně jsou řešeny bakalářské práce *Extrakce metadat z vědeckých článků* Tomáše Lokaje a *Převod vědeckých článků na text* Jiřího Matičky a také diplomá práce *Automatické navrhování klíčových slov* Tomáše Strachoty. Uvedené práce budou také použity v systému ReReSearch a jsou tedy uvedeny mezi moduly.

# Literatura


[1] Arms, W. Y.: *Digital Libraries*. The MIT Press, 2000, ISBN 0-262-01880-8.

[2] Bartošek, M.: Digitální knihovny [online]. In *Datakon 2001*, 2001 [cit. 17. dubna 2010].
Dostupné na URL: http://www.ics.muni.cz/mba/dl-datakon01.pdf

[3] Candela, L.; Castelli, D.; Ferro, N.; aj.: *The DELOS Digital Library Reference Model - Foundations for Digital Libraries* [online]. February 2008 [cit. 17. dubna 2010].
Dostupné na URL: http://www.delos.info/files/pdf/ReferenceModel/DELOS_DLReferenceModel_0.98.pdf

[4] WWW stránky: O službě Google Scholar [online]. 2010 [cit. 28. dubna 2010].
Dostupné na URL: http://scholar.google.com/scholar/about.html

[5] WWW stránky: Google Press Center: Product Descriptions [online]. 2010 [cit. 28. dubna 2010].
Dostupné na URL: http://www.google.com/press/descriptions.html

[6] Bartošek, M.: Nástroje Google 2: Google Scholar. In *Zpravodaj ÚVT MU*, 2008, s. 12–16.

[7] WWW stránky: What is ACM? [online]. 2010 [cit. 29. dubna 2010].
Dostupné na URL: http://www.acm.org/about

[8] WWW stránky: ACM Fact Sheet [online]. 2010 [cit. 29. dubna 2010].
Dostupné na URL: http://www.acm.org/about/fact_sheet

[9] WWW stránky: About IEEE [online]. 2010 [cit. 29. dubna 2010].
Dostupné na URL: http://www.ieee.org/about/index.html

[10] WWW stránky: IEEE at a Glance [online]. 2010 [cit. 29. dubna 2010].
Dostupné na URL: http://www.ieee.org/about/at_a_glance.html

[11] WWW stránky: About CiteSeer [online]. 2010 [cit. 31. dubna 2010].
Dostupné na URL: http://citeseer.ist.psu.edu/citeseer.html

[12] WWW stránky: About CiteSeerX [online]. 2010 [cit. 31. dubna 2010].
Dostupné na URL: http://citeseerx.ist.psu.edu/about/site

[13] Ley, M.; Reuther, P.: Maintaining an Online Bibliographical Database: The Problem of Data Quality. In *EGC*, 2006, s. 5–10.

[14] WWW stránky: About the ISI Web of Knowledge [online]. 2010 [cit. 9. května 2010]. Dostupné na URL: http://wok.mimas.ac.uk/about/

[15] Smrž, P.: ReReSearch [online]. 2009 [cit. 28. dubna 2010], interní dokumentace projektu ReReSearch.
Dostupné na URL:
https://merlin.fit.vutbr.cz/nlp-wiki/index.php/ReReSearch

[16] Smrž, P.; Nováček, V.: Ontology Acquisition for Automatic Building of Scientific Portals. In *SOFSEM 2006: Theory and Practice of Computer Science: 32nd Conference on Current Trends in Theory and Practice of Computer Science*, Springer Verlag, 2006, ISBN 3-540-31198-X, s. 493–500.

[17] Horák, J.: *Řídicí systém generátoru vědeckých webových portálů*. Bakalářská práce, FIT VUT v Brně, 2008.

[18] Lokaj, T.: Převod publikací do textového tvaru a jejich indexování [online]. 2010 [cit. 5. května 2010], interní dokumentace projektu ReReSearch.
Dostupné na URL:
https://merlin.fit.vutbr.cz/nlp-wiki/index.php/Rrs_article2meta

[19] Strachota, T.: Automatické navrhování klíčových slov [online]. 2010 [cit. 5. května 2010], interní dokumentace projektu ReReSearch.
Dostupné na URL:
https://merlin.fit.vutbr.cz/nlp-wiki/index.php/Rrs_keywords

[20] Zelinka, M.: Začlenění dat citačních databází [online]. 2010 [cit. 6. května 2010], interní dokumentace projektu ReReSearch.
Dostupné na URL:
https://merlin.fit.vutbr.cz/nlp-wiki/index.php/Rrs_citdb_integration

[21] Homoliak, I.: Extrakce a zpracování informací o projektech [online]. 2010 [cit. 6. května 2010], interní dokumentace projektu ReReSearch.
Dostupné na URL:
https://merlin.fit.vutbr.cz/nlp-wiki/index.php/Rrs_eu_projects

[22] Heller, S.: Vyhledání osobních stránek autorů a seznamu jejich publikací [online]. 2010 [cit. 6. května 2010], interní dokumentace projektu ReReSearch.
Dostupné na URL:
https://merlin.fit.vutbr.cz/nlp-wiki/index.php/Rrs_homepagesearch

[23] Heller, S.: Vyhledávání a zpracování výzkumných zpráv projektů [online]. 2010 [cit. 7. května 2010], interní dokumentace projektu ReReSearch.
Dostupné na URL:
https://merlin.fit.vutbr.cz/nlp-wiki/index.php/Rrs_deliverables

[24] Novák, J.: Sémantická blízkosť termov [online]. 2010 [cit. 7. května 2010], interní dokumentace projektu ReReSearch.
Dostupné na URL:
https://merlin.fit.vutbr.cz/nlp-wiki/index.php/Blizkost_termu

[25] Horák, J.: Řídicí systém GVP [online]. 2009 [8. května 2010], interní dokumentace projektu ReReSearch.
Dostupné na URL: https://merlin.fit.vutbr.cz/nlp-wiki/index.php/GVP:%C5%98%C3%ADdic%C3%AD_syst%C3%A9m

[26] Bartoš, P.; Angelov, M.: Zjednotenie návrhu databázy a dotazovací systém [online]. 2009 [cit. 9. května 2010], interní dokumentace projektu ReReSearch.
Dostupné na URL: https://merlin.fit.vutbr.cz/nlp-wiki/index.php/GVP:Zjednotenie_n%C3%A1vrhu_datab%C3%A1zy_a_dotazovac%C3%AD_syst%C3%A9m

[27] WWW stránky: XML [online]. 2010 [cit. 22. dubna 2010].
Dostupné na URL: http://www.w3.org/standards/xml/core

[28] WWW stránky: Schema intro [online]. 2010 [cit. 23. dubna 2010].
Dostupné na URL: http://www.w3schools.com/schema/schema_intro.asp

[29] Wang, X.; Yin, Y. L.; Yu, H.: Finding collisions in the full SHA-1. In *Crypto'05*, 2005.

[30] Patashnik, O.: BibTEX yesterday, today, and tomorrow [online]. In *Proceedings of the 2003 Annual Meeting*, 2003 [cit. 3. května 2010], str. 24:25–30.
Dostupné na URL:
http://www.tug.org/TUGboat/Articles/tb24-1/patashnik.pdf

[31] Parashnik, O.: *BibTEXing* [online]. 1988 [cit. 3. května 2010].
Dostupné na URL:
http://mirror.ctan.org/biblio/bibtex/contrib/doc/btxdoc.pdf

[32] Brenno, L. P.; Previtali, L.; Lurati, B.; aj.: BIBTEXML: An XML Representation of BIBTEX. In *In Poster Proceedings of the Tenth International World Wide Web Conference*, ACM Press, 2001, s. 64–65.

[33] Fielding, R.; Gettys, J.: RFC 2616: Hypertext Transfer Protocol – HTTP/1.1 [online]. June 1999 [cit. 3. května 2010].
Dostupné na URL: http://tools.ietf.org/html/rfc2616

[34] Nottingham, M.; Modul, J.: RFC 4229: HTTP Header Field Registrations [online]. December 2005 [cit. 3. května 2010].
Dostupné na URL: http://tools.ietf.org/html/rfc4229

[35] WWW stránky: PostgreSQL 8.4.3 Documentation [online]. 2009 [cit. 14. dubna 2010].
Dostupné na URL: http://www.postgresql.org/docs/8.4/static/index.html

[36] Douglas, K.; Douglas, S.: *PostgreSQL: The comprehensive guide to building, programming, and administering PostgreSQL databases, Second Edition*. Sams Publishing, 2005, ISBN 0-672-32756-2.

[37] Dlouhý, I.: Články odkazované na CiteSeerX [online]. 2010 [cit. 10. května 2010], interní dokumentace projektu ReReSearch.
Dostupné na URL: https://merlin.fit.vutbr.cz/nlp-wiki/index.php/Testovac%C3%AD_%C4%8Dl%C3%A1nky

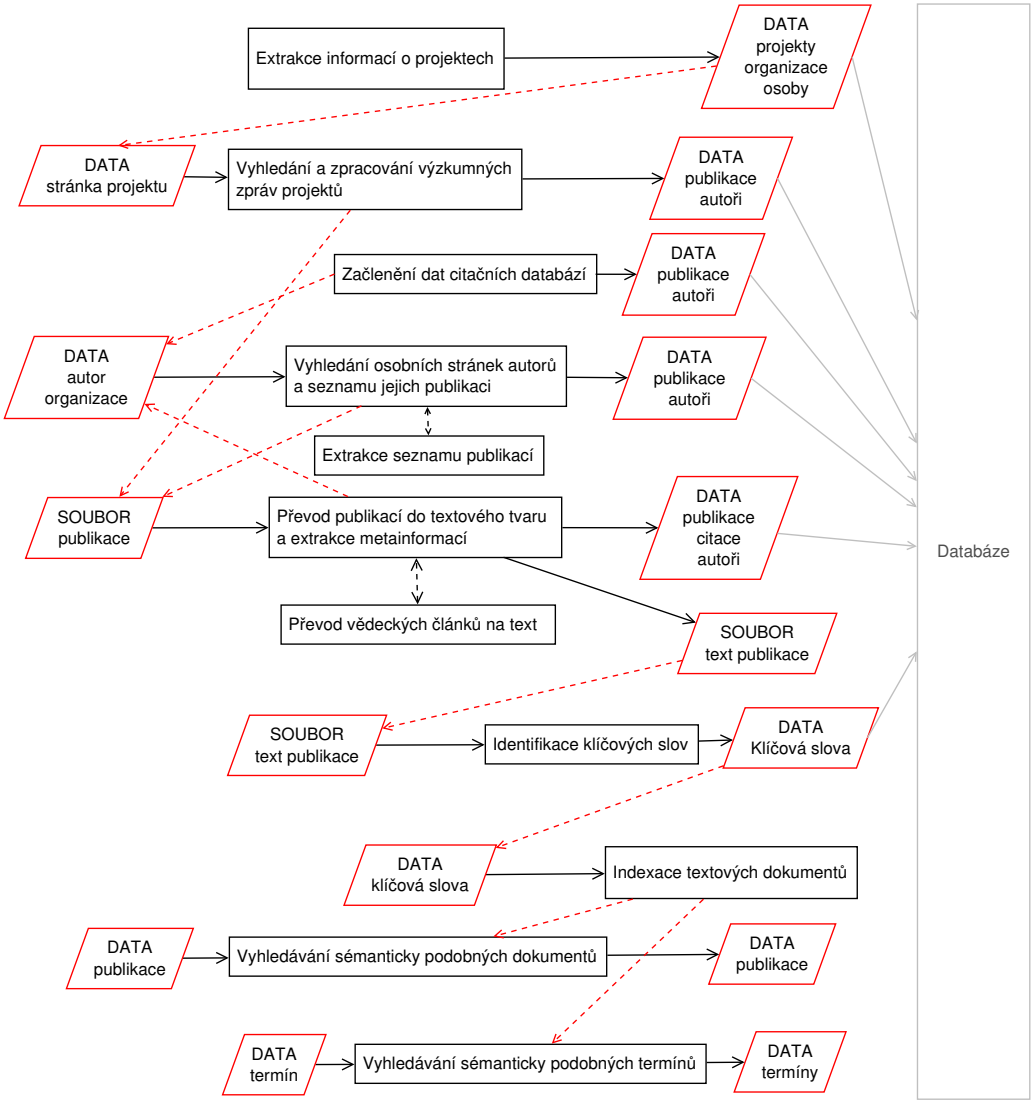
# Příloha A

# Schéma databáze pro data

Vloženo jako samostatná příloha.

# Příloha B

# Propojení modulů




Obrázek B.1: Přehled modulů

# Příloha C

# Databázové typy

## Typy publikací

Typy publikací jsou vysvětleny v tabulce C.1.

| | |
|---|---|
| `article` | článek z časopisu, novin, sborníku |
| `book` | kniha s uvedeným vydavatelem |
| `booklet` | kniha bez vydavatele nebo sponzorující instituce |
| `conference` | viz. inproceedings |
| `inbook` | často nepoznamenaná část knihy, kapitola, rozsah stránek |
| `incollection` | pojmenovaná část knihy |
| `inproceedings` | článek ve sborníku konference |
| `manual` | technická dokumentace |
| `mastersthesis` | diplomová práce |
| `misc` | ostatní |
| `phdthesis` | disertační práce |
| `proceedings` | sborník ke konferenci |
| `techreport` | technická zpráva publikovaná školou nebo jinou institucí, často číslovaná |
| `tunpublished` | publikace s názvem a autorem, nepublikovaná |

Tabulka C.1: Typy publikací

## Typy událostí

Typy událostí jsou vysvětleny v tabulce C.2.

| | |
|---|---|
| `award` | ocenění |
| `ceremony` | oficiální setkání k oslavě nějáké události |
| `competition` | soutěž |
| `concert` | koncert |
| `conference` | konference |
| `convention` | formální setkání |
| `course` | kurz, školní výukový program |
| `defense` | obhajoba titulu |
| `discussion` | diskuze, konverzace nebo debata okolo tématu |
| `exhibition` | výstava, expozice |
| `fair` | veletrh, pravidelný trh |
| `forum` | typ diskuze s porotou a velkou učastí publika |
| `launch` | uvedení produktu, technologie |
| `lecture` | přednáška na určité téma za účelem vzdělávání |
| `meeting` | setkání, schůze |
| `misc` | ostatní |
| `open day` | den otevřených dveří |
| `screening` | promítání filmu |
| `seminar` | seminář, školení |
| `social` | společenská událost |
| `symposium` | setkání, konference za účelem diskuze tématu, účastníci mají příspěvky |
| `webinar` | interaktivní seminář online, prezentace, posluchači se mohou účastnit |
| `workshop` | intenzivní vzdělávací kurz pro malou skupinu |

Tabulka C.2: Typy událostí

# Příloha D

# Obsah přiloženého média

| | |
|---|---|
| `doc/src` | zdrojové kódy technické zprávy |
| `doc/dlouhy-bp.pdf` | elektronická verze technické zprávy |
| `doc/dlouhy-bp-plakat.pdf` | plakát |
| `doc/dlouhy-bp-prilohaA.pdf` | externí příloha A |
| `reresearch` | adresář se zdrojovými kódy |
| `README` | popis obsahu média |