Západočeská univerzita v Plzni
Fakulta aplikovaných věd
Katedra informatiky a výpočetní techniky

Bakalářská práce

Import dat ze služby Scopus do formátu XML

Plzeň, 2012 Rudolf Augusta

Prohlašuji, že jsem diplomovou práci vypracoval samostatně a výhradně s použitím citovaných pramenů.

V Plzni dne 14. 4. 2012

Rudolf Augusta

Abstract

Import of data from Scopus to XML

Scientists and academics from the whole world are trying to invent something new. They are looking for inspiration in scientific work of their colleagues. Information regarding these works are stored in bibliographic databases. One of these databases is Scopus created by Elsevier.

Target of this work is to create an application that will retrieve information from Web application Scopus. The information obtained also will save to a file in XML format.

Import dat ze služby Scopus do formatu XML

Vědci a akademičtí pracovníci z celého světa se snaží vymyslet něco nového. Inspiraci hledají ve vědeckých prácích svých kolegů. Informace týkající se těchto prací jsou uchovávány v bibliografických databázích. Jednou takovou databází je i Scopus od firmy Elsevier.

Cílem této práce je vytvořit aplikaci, která bude získávat informace z webové aplikace Scopus. Získané informace dále uloží do souboru ve formátu XML.

Obsah

1 Úvod ... 1

2 Scopus ... 2

2.1 Vyhledávání dokumentů ... 2

2.1.1 Vyhledávací formulář ... 2

2.1.2 Vyhledané dokumenty ... 3

2.1.3 Detail dokumentu ... 3

2.1.4 Detail zdroje ... 4

2.1.5 Dokumenty citující daný dokument ... 4

2.2 Vyhledávání autorů ... 4

2.2.1 Vyhledávací formulář ... 4

2.2.2 Vyhledaní autoři ... 5

2.2.3 Detail autora ... 5

2.3 Vyhledávání afilací ... 5

2.3.1 Vyhledané afilace ... 6

2.3.2 Detail afilace ... 6

2.4 Pokročilé vyhledávání ... 6

2.5 Export dat ... 7

2.6 Omezení ... 9

3 Technologie ... 10

3.1 HTTP ... 10

3.2 HTML ... 12

3.2.1 Struktura HTML dokumentu ... 13

3.3 HTML Parser ... 15

3.4 XML ... 16

3.5 JAXB ... 17

3.6 StAX ... 17

4 Implementace ... 18

4.1 Použité technologie ... 18

4.2 Architektura ... 18

4.2.1 Prezentační vrstva ... 19

4.2.2 Aplikační vrstva ... 19

4.2.3 Datová vrstva ... 20

4.3 Získávání informací ... 20

4.3.1 Formulář ... 20
4.3.2 Vyhledané dokumenty ... 20
4.3.3 Detail dokumentu ... 21
4.3.4 Detail autora ... 22
4.3.5 Detail zdroje ... 22
4.3.6 Dokumenty citující daný dokument ... 22
4.4 Získání další generace ... 22
4.5 Ukládání výsledků ... 23
4.5.1 Formát ukládání ... 23
5 Testování ... 26
6 Závěr ... 27
Přehled zkratek ... 28
Literatura ... 29
Příloha A: Uživatelský manuál ... 30
A.1 Spuštění programu ... 31
A.2 Popis GUI ... 31
A.2.1 Nastavení hledání a omezení ... 32
A.2.2 Konzole ... 32
A.2.3 Stavový řádek ... 32
Příloha B: Výstupní XML ... 33

1 Úvod

Věda se stala součástí našich životů, přestože je stále co objevovat. Vědci a akademičtí pracovníci se při výzkumu často inspirují z vědeckých prací svých kolegů. Dokumentů pro inspiraci je většinou velké množství. Přestože, některé práce nelze volně získat, jsou i práce vycházející na konferencích, v odborných časopisech atd. Také na internetu lze nalézt některé z těchto dokumentů. Pro vědce je složité takové práce najít. Proto se vytvořili bibliografické databáze, které slouží ke shromáždění informací týkající se těchto prací. V databázích se často vyskytují i odkazy na celé dokumenty v elektronické podobě, ale u některých dokumentů jsou uvedeny pouze základní informace, jako je například název, rok vydání, autoři, odkaz na práce, které daný dokument citují atd.

Mezi tyto bibliografické databáze patří projekt Scopus [1] od firmy Elsevier [2]. Tato databáze obsahuje stále velké množství dokumentů. Proto by bylo dobré mít vlastní databázi, ve které budou pouze vybrané dokumenty. Problém při vytváření vlastní databáze nastává, při získávání a uchovávání informací. Z tohoto důvodu vznikla tato bakalářská práce.

Cílem bakalářské práce je vytvořit aplikaci, která bude automatizovaně získávat informace o dokumentech z webové aplikace Scopus. Získané informace dále uloží do souboru ve formátu XML (*Extensible Markup Language*).

2 Scopus

Scopus je projektem nakladatelské firmy Elsevier, který byl vytvořen v roce 2004. Je to největší abstraktová a citační databáze na světě. Obsahuje přes 19000 titulů od více než 5000 mezinárodních vydavatelů. Scopus je podporou akademickým a vědeckým pracovníkům z oblasti vědy, techniky, lékařství a sociálních věd. Poslední dobou se zabývá také oblastí umění a humanitních věd. K jeho používání je zapotřebí koupit si licenci.

Scopus umožňuje čtyři druhy vyhledávání:

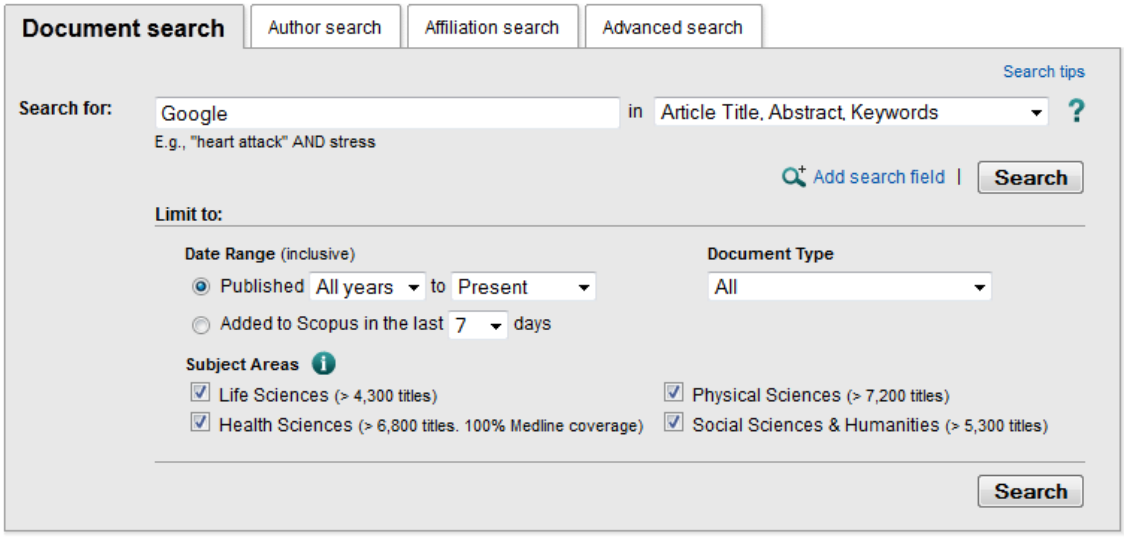
- vyhledávání dokumentů (Document search)
- vyhledávání autorů (Author search)
- vyhledávání afilací (Affiliation search)
- pokročilé vyhledávání (Advanced search)

2.1 Vyhledávání dokumentů

2.1.1 Vyhledávací formulář

Při zadání URL adresy www.scopus.com do webového prohlížeče se uživateli zobrazí vyhledávací formulář (viz. Obr. 2.1).

V tomto formuláři lze nastavit, co chce uživatel hledat: *Search for*. Dále lze nastavit, v jaké části se hledaný řetězec má vyskytovat: *in*, jestli řetězec hledáme v abstraktu, názvu dokumentu, nebo hledaný řetězec je jméno autora, nebo název konference, na které byl dokument vydán atd. Také lze nastavit *Published*, to slouží k omezení, aby se zobrazili dokumenty z určitého období. V neposlední řadě můžeme nastavit pomocí *Document type*, jakého typu hledané dokumenty mají být. A poslední nastavení je v jakém odvětví výzkumu chceme hledat.



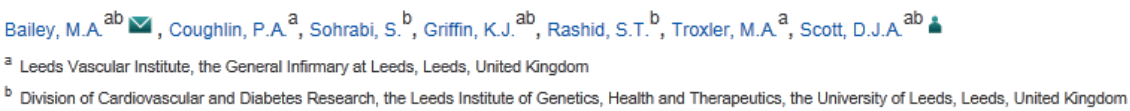
Obr. 2.1: Formulář pro vyhledávání dokumentů.

2.1.2 Vyhledané dokumenty

Po zmáčknutí tlačítka *Search* je uživatel přesměrován na stránku s vyhledanými dokumenty. V horní části této stránky, je informace o požadavku vyhledávání. V levé části lze nastavit omezení podobná těm na formuláři. V pravé časti, jsou informace o dokumentech. Každý dokument má uveden název, autory, rok vydání, informace o zdroji a počet citujících dokumentů, které daný dokument citují. Také lze pomocí odkazů na této stránce přejít k dalším informacím

2.1.3 Detail dokumentu

Kliknutím na název dokumentu se uživatel dostane na stránku s detailními informacemi o daném dokumentu. Na této stránce jsou informace o zdroji dokumentu, název, autoři a afiliace, kterou autoři měli, když dokument psaly. Jednotlivé afilace jsou k autorům přiřazeny pomocí indexů (viz Obr. 2.2). Dále zde uživatel může nalézt abstrakt a reference na dokumenty, které dokument citoval.



Obr. 2.2: Ukázka přiřazení afiliací k autorům.

2.1.4 Detail zdroje

Po kliknutí na název zdroje se uživatel dostane na stránku s informacemi o zdroji daného dokumentu. Jsou zde informace o vydavateli, jakou částí vědy se zdroj zabývá a od jakého roku se vydává.

2.1.5 Dokumenty citující daný dokument

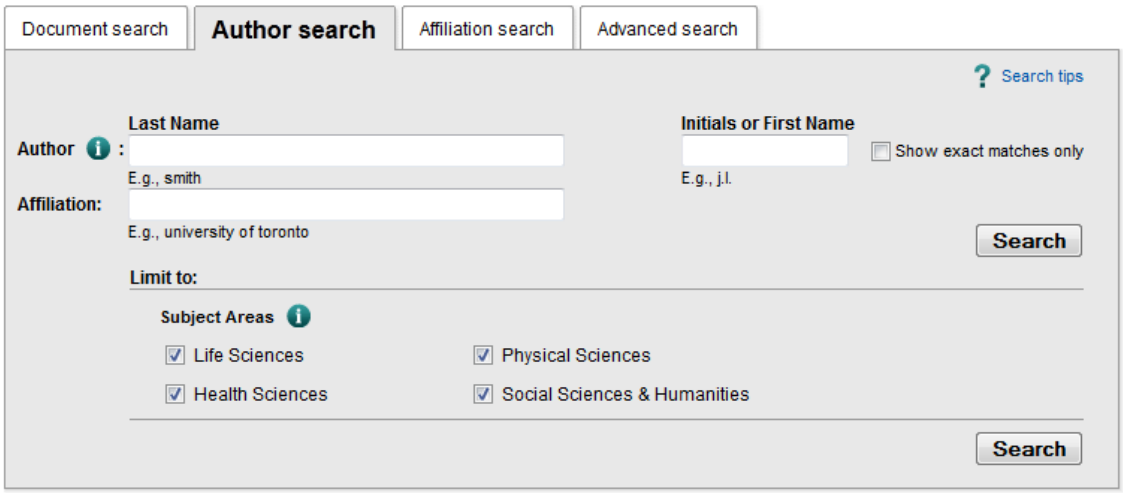
Poslední z odkazů ze stránky s vyhledanými dokumenty je odkaz na stránku s citujícími dokumenty. Tato stránka má stejnou strukturu jako stránka s vyhledanými dokumenty (viz kapitola 2.1.2).

2.2 Vyhledávání autorů

2.2.1 Vyhledávací formulář

Po přepnutí na lištu *Author search* se uživateli zobrazí vyhledávací formulář (viz. Obr. 2.3), sloužící k vyhledávání autorů podle jména a příjmení.

Pro upřesnění hledání lze zadat autorovu afilaci (členství – nejčastěji univerzita) a odvětví vědy, kterým se autor zabývá.



Obr. 2.3: Formulář pro vyhledávání autorů.

2.2.2 Vyhledaní autoři

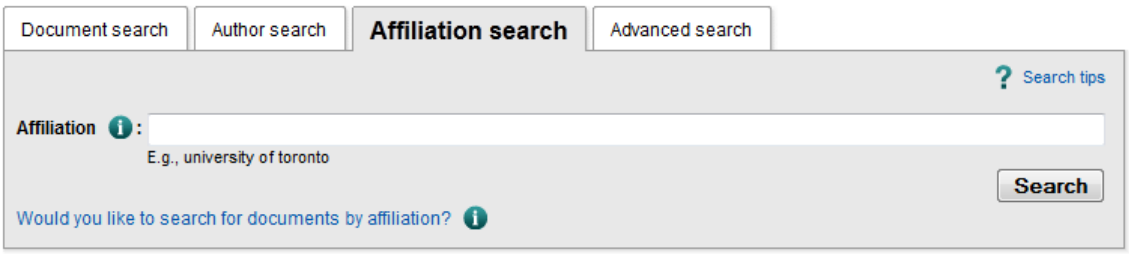
Po kliknutí na tlačítko *Search*, server uživatele přesměruje na stránku s vyhledanými autory. V levé části jsou omezení. Na pravé části jsou informace týkající se autorů. Každý autor má uvedené příjmení a první písmeno jména (popřípadě celé jméno), počet vydaných dokumentů, odvětví vědy, kterým se zabývá, město a stát týkající se jeho afilace.

2.2.3 Detail autora

V levé části stránky jsou detailní informace o autorovy. Tyto informace jsou rozdělený do tří skupy. První ze skupiny jsou osobní informace. Sem patří jméno, jiné formáty jména uvedená kvůli vyhledávání autora, ID a afilace. Další skupinou je výzkum. Zde můžeme nalézt počet vydaných dokumentů, počet dokumentů, na které autor odkazuje, počet citací, kolikrát byl autor citován, h index, oblast vědy, kterou se autor zabývá atd. Poslední skupinou je historie. V této části je rozsah roků, od vydání prvního autorova dokumentu do posledního. Dále jsou zde zdroje, ze kterých se čerpalo při výpisu roků.

2.3 Vyhledávání afilací

Dalším vyhledáváním, které Scopus podporuje, je vyhledávání afilací (viz. Obr 2.4). Afilace je společnost, pod kterou autoři vydávají vědecké práce. Například Západočeská univerzita v Plzni.



Obr. 2.4: Formulář pro vyhledávání afilací.

2.3.1 Vyhledané afilace

Na této stránce jsou, v levé části opět omezení. V pravé části jsou výsledky hledání s informacemi: název afilace, počet vydaných dokumentů, město, ve kterém se afilace vyskytuje, a stát.

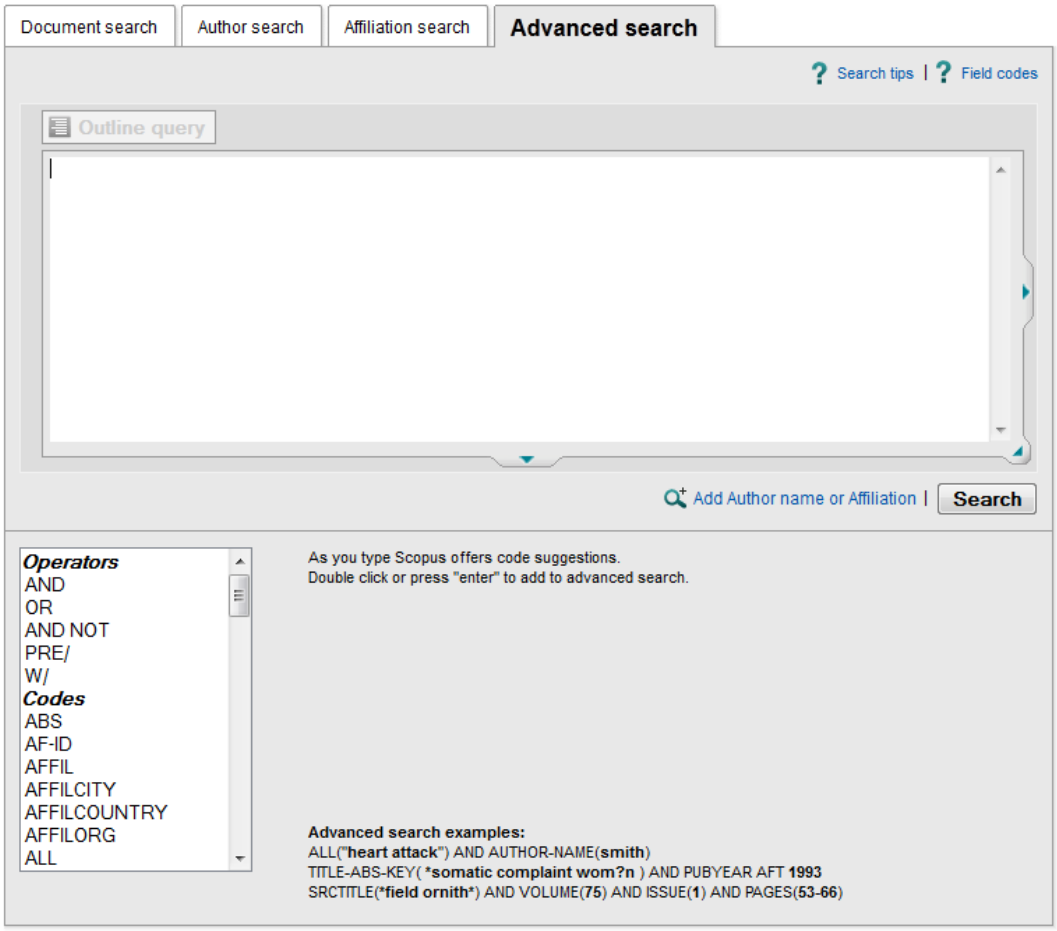
2.3.2 Detail afilace

V levé části jsou informace o afilaci rozděleny do tří částí. První části je název afilace, ID, adresa a formáty jména, pod kterými ji lze vyhledávat. V druhé části jsou informace o výzkumu. Je zde uvedeno, kolik dokumentů vydali autoři v době působení na dané afilaci. Také je zde uveden počet autorů atd. V poslední části jsou afilace, se kterými daná afilace spolupracuje.

V pravé části je graf, ve kterém je vidět procentuální zastoupení vydaných dokumentů přiřazených do jednotlivých odvětví vědy na afilaci.

2.4 Pokročilé vyhledávání

Poslední možností vyhledávání je pokročilé vyhledávání (viz. Obr 2.5). Zde si uživatel může pomocí booleovských operátorů vytvořit řetězec s omezením. Na příkladu (viz Obr. 2.5) je nastavené hledání řetězce *google* v názvech, abstraktech a klíčových slovech dokumentů. Toto hledání je navíc omezené datem vydání dokumentů od roku 1700 do roku 2010. Vyhledané výsledky mají stejnou strukturu jako výsledky vyhledané pomocí formuláře pro vyhledávání dokumentů.



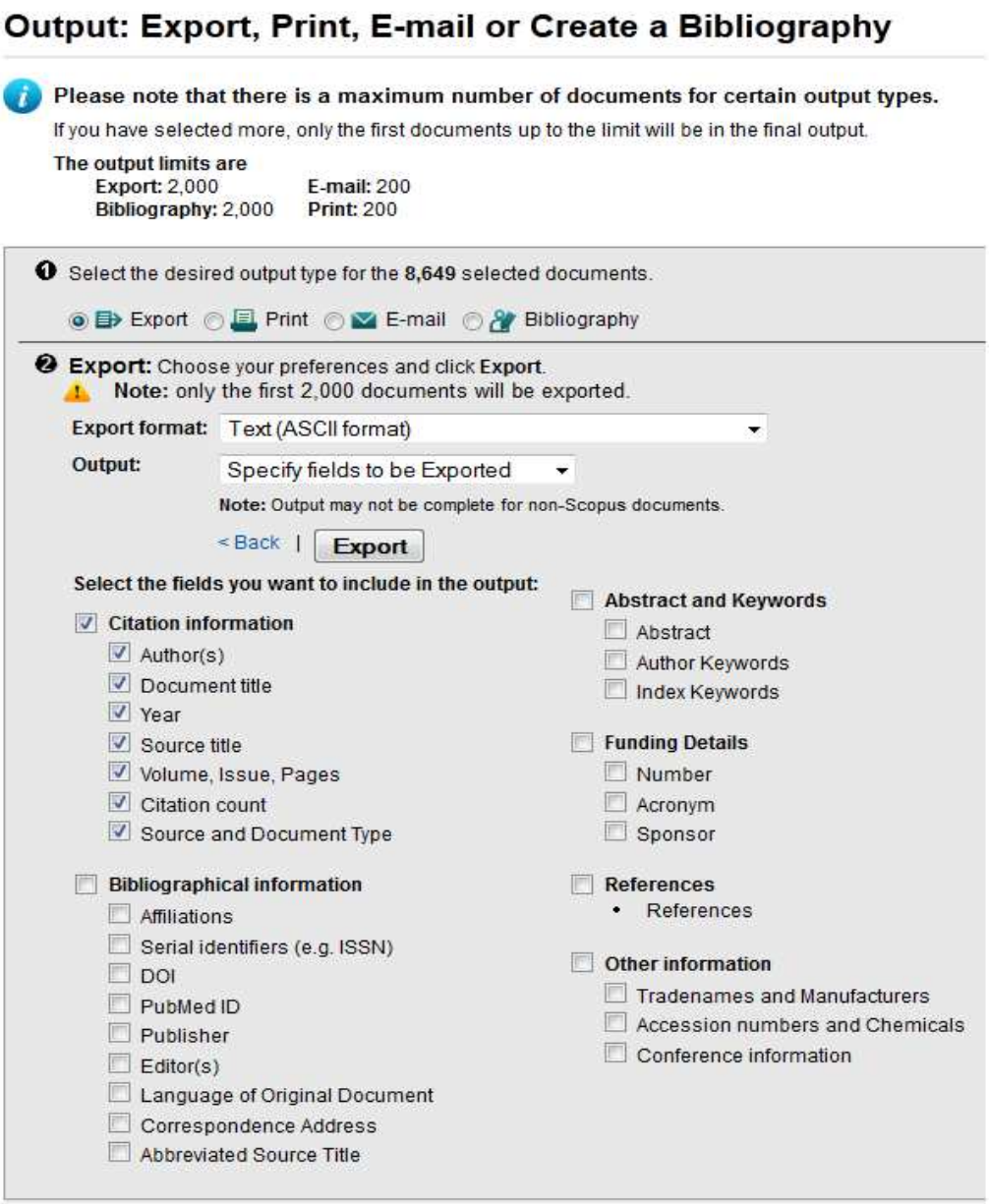
Obr. 2.5: Formulář pro pokročilé vyhledávání.

2.5 Export dat

Scopus umožňuje vyexportovat vyhledané dokumenty do pěti formátů:

- Text (ASCII format)
- RefWork direct export
- RIS format (Reference Manager, ProCite, EndNote)
- BibTex
- Comma separated file, .csv (e.g. Excel)

Dále je zde možnost vybrat, jaké informace mají být ve výstupním formátu (viz Obr. 2.6).

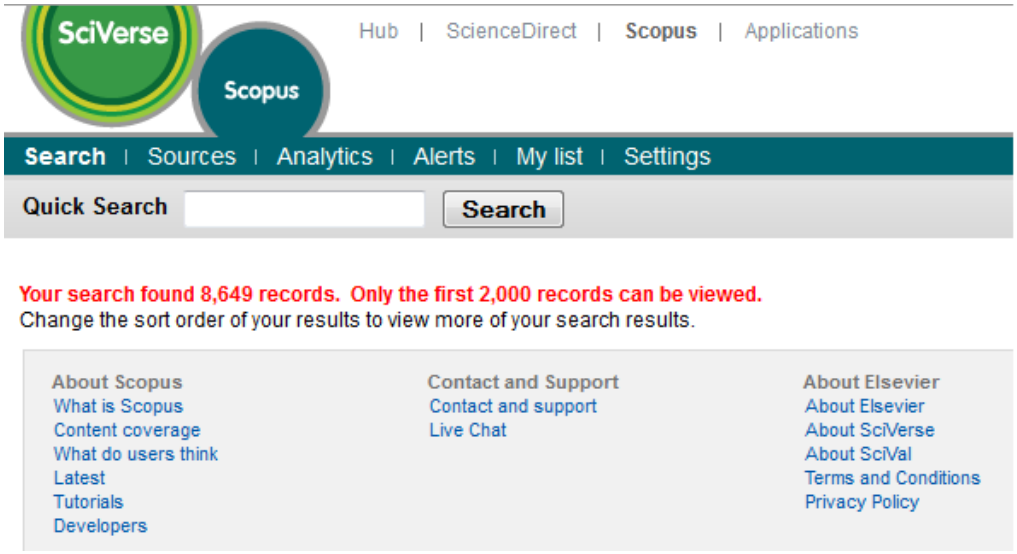


Obr. 2.6: Možnost exportu dat.

Žádná z těchto informací, které se mohou exportovat do jednotlivých výstupních formátů, neidentifikují dokument, autora nebo afilaci jednoznačně. Dále nelze exportovat do výstupních formátů informace, jaké dokumenty daný dokument citují. Tyto dvě informace jsou pro nás celkem důležité.

2.6 Omezení

Služba Scopus má jedno velmi nepříjemné omezení. Lze zobrazit nebo exportovat pouze prvních 2000 vyhledaných informací. Jakmile se chce uživatel podívat na informaci na pozici 2001 Scopus nám zobrazí hlášku: „Your search found X records. Only first 2000 recods can be viewed.“ (viz. Obr. 2.7) kde X je číslo větší jak 2000.



Obr. 2.7: Omezení webové služby Scopus.

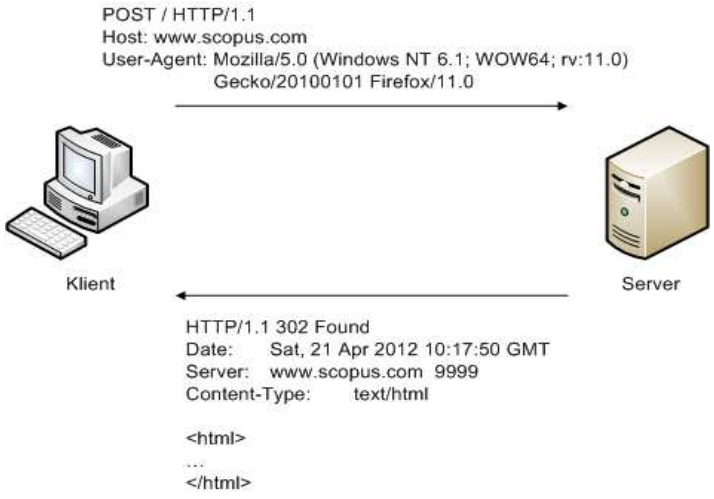
3 Technologie

3.1 HTTP

HTTP (*HyperText Transfer Protocol*) [3] je internetový protokol, původně sloužící k přenosu dokumentů ve formátu HTML (*HyperText Markup Language*) [4]. K tomuto přenosu je nejčastěji použit port TCP/80. V současné době je nejpoužívanější HTTP protokol verze 1.1.

Protokol funguje na bázi dotaz – odpověď (viz. Obr. 3.1). Veškerou komunikaci tedy začínají klienti (nejčastěji webové prohlížeče) odesláním dotazu na server. Dotaz má podobu čistého textu a jsou v něm uvedeny informace o daném prohlížeči. Server poté odpoví opět formou textu, zda daný dokument nalezl, jakého je formátu atd. Za textem se již nachází data samotného dokumentu.

Tento protokol je bezestavový. To znamená, že si server neuchovává žádné spojení s klientem ani informace o něm. Například, když pošle klient dotaz, server odpoví. Při poslání dalšího dotazu se jedná o dotaz nezávislý na předchozím. Díky tomu může server obsluhovat velké množství klientů s minimálními paměťovými nároky.



Obr. 3.1: Klient - server.

Metody HTTP protokolu verze 1.1:

- GET
- POST
- HEAD
- PUT
- DELETE
- TRACE
- PATCH
- OPTIONS
- CONNECT

GET

Metoda GET používá URI (*Uniform Resource Identifier*) k získávání webových stránek nebo jejich částí. URI slouží k jednoznačné identifikaci jednotlivých objektů na WWW (*World Wide Web*) serverech.

Při předávání proměnných z formuláře jsou jednotlivé proměnné oddělené ampersandy, kódují se za otazník přímo do URI adresy a jsou ve formě *proměnná=hodnota.* V následujícím příkladu se odesílají proměnné: *eid* (*elektronický identifikátor*), která slouží k identifikaci dokumentu a *origin*.

Příklad:

http://www.scopus.com/record/display.url?eid=2-s2.0-84856884078&origin=resultslist

POST

Metoda POST, také jako metoda GET, používá URI k získání webových stránek nebo jejich částí. Oproti metodě GET se proměnné z formuláře nezobrazují v URI adrese, ale posílají se jako součást HTTP požadavku.

HEAD

Tato metoda je téměř shodná s metodou GET. Jediný rozdíl u těchto metod je, že HEAD neposílá celý HTML dokument, ale pouze informace o jeho velikosti, typu, datum poslední změny atd.

PUT

Tato metoda slouží k nahrání dat na server. V dotazu se určí, kam se data mají nahrát a metoda PUT je vytvoří. K tomu jsou potřeba určitá oprávnění.

DELETE

Metoda DELETE smaže data ze serveru. Stejně jako u metody PUT jsou zapotřebí určitá oprávnění.

PATCH

Tato metoda slouží k upravení dat na serveru. Umožňuje poslat jen rozdílový dokument – popis změn, které je třeba provést s verzí, již server aktuálně nabízí.

TRACE

Pomocí této metody server odešle kopii požadavku zpět k odesílateli. Ten může pomocí této kopie zjistit, co na požadavku mění servery, kterými požadavek prošel na cestě k příjemci.

OPTIONS

Metoda OPTIONS se dotazuje na serverem podporované metody.

CONNECT

Spojí se přes uvedený port a vytvoří TCP/IP trvalé propojení. Používá se v HTTPS (*Hypertext Transfer Protocol Secure*) při průchodu skrze proxy.

3.2 HTML

HTML je značkovací jazyk pro vytváření stránek v systému WWW. Umožňuje publikaci dokumentů na internetu. Tento jazyk byl vyvinut z univerzálního jazyka SGML (*Standard Generalized Markup Language*) [5]. Vývoj jazyka HTML byl ovlivněn vývojem webových prohlížečů. Mezi nejrozšířenější patří HTML verze 4.0.

Jazyk HTML vytvořil Tim Berners-Lee v roce 1991, při práci na propojeném informačním systému pro společnost CERN (*Centre Européenne pour la Recherche Nucléaire*, Evropské centrum jaderného výzkumu). Informační systém měl vědcům umožnit, rozšíření výsledků výzkumu do celého světa. První verze jazyka HTML byla popsána v dokumentu HTML tags[6]. Umožňovala text rozčlenit do několika logických úrovní, použít několik druhů zvýraznění textu a zařadit do textu odkazy a obrázky.

Jazyk je charakterizován pevně určenou množinou značek (tzv. tagů), které tvoří HTML elementy. Tyto elementy mohou být párové nebo nepárové. Párové se skládají z otevírací a uzavírací značky. Mezi značkami se uzavírá část textového dokumentu nebo další vnořené značky. Nepárové nemají žádný obsah a nepoužívají koncové značky. Jednotlivé značky určují sémantiku obsaženého textu. Názvy značek se uzavírají do špičatých závorek např.: <p>, je otevírací značka nového odstavce. Otevírací značky mohou mít přiřazené atributy. Ty se uvádí ve formě `atribut="hodnota"`, za názvem značky, a jsou oddělené mezerami např.: `<p atribut1="hodnota1" atribut2="hodnota2" … >`. Uzavírací značky odlišujeme od otevíracích tím, že před názvem značky je uveden znak lomeno.

3.2.1 Struktura HTML dokumentu

HTML dokument má předepsanou strukturu (viz. Obr. 3.2):

- Deklarace DTD (*Document Type Definition*)
- Kořenový element
- Hlavička dokumentu
- Tělo dokumentu

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Final//CZ">
<html>
    <head>
        <!-- Obsah hlavičky -->
    </head>
    <body>
        <!-- Obsah těla -->
    </body>
</html>
```

Obr. 3.2: Struktura HTML dokumentu.

Deklarace DTD

Tato deklarace určuje jakého typu daný dokument je. Píše se na začátek dokumentu před otevírací značku <html>. Deklarace je uvedena direktivou <!DOCTYPE.

Na příkladu (viz. Obr. 3.2) je vidět, že daný dokument je typu HTML. Verze použitého jazyka HTML je 4.0 a dokument obsahuje české znaky.

Kořenový element

Dokument uvozuje otevírací značka <html> a ukončuje uzavírací párová značka </html>. Veškerý další obsah musí být uzavřen uvnitř tohoto elementu. Tyto značky není povinné uvádět. Většina nových prohlížečů si je při zobrazování stránky doplní. Pokud má být dokument v souladu s normou, značky je nutné uvádět.

Hlavička dokumentu

Hlavička dokumentu je uvedena otevírací značkou <head>. Údaje zde uvedené se v samotné stránce nezobrazují. Jediná zobrazená informace definovaná v hlavičce je titulek <title>. Jedná se o název stránky, který se zobrazí v liště webového prohlížeče nebo v historii. Další důležité značky, které lze nalézt v hlavičce jsou link, meta a script.

Značka <link> propojuje HTML dokument s jiným souborem. Nejčastěji se toto propojení používá pro načtení externího stylu CSS:

```
<link rel="stylesheet" type="text/css" href="styl.css">
```

Informace o dokumentu, metadata jsou ve značce <meta>. Pomocí této značky se nastavuje jazyk dokumentu a kódování. Díky tomuto nastavení se správně zobrazují české znaky. Nastavení kódování windows-1250 a českého jazyka:

```
<meta http-equiv="Content-Type" content="text/html;
charset=windows-1250" />

<meta http-equiv="content-language" content="cs" />
```

Značka <script> připojí ke stránce skript, obvykle JavaScript. Ukázka:

```
<script language="JavaScript" type="text/javascript"
src="skript.js">
```

Tělo dokumentu

Tělo dokumentu uvozuje otevírací značka <body>. Tato část dokumentu slouží k tomu, aby zobrazila samotný obsah stránky, tedy vše co je vidět ve webovém prohlížeči.

K tomu, aby dokument po zobrazení v prohlížeči měl určitou strukturu, se využívá zanořování jednotlivých HTML elementů. Většina z nich, má jako svého potomka textový element. Ten může být prázdný, nebo může obsahovat text. Tento text se zobrazuje jako obsah stránky při načtení HTML dokumentu v prohlížeči. Dále také mohou mít jako potomka jakýkoliv jiný HTML element, který od svého rodiče dědí styl, jakým se zobrazí.

3.3 HTML Parser

HTML stránky, které klient získá od serveru, jsou velmi obsáhlé. Z tohoto důvodu je zapotřebí identifikovat pouze potřebné informace a ostatní odfiltrovat. Toho lze dosáhnout velmi složitě, protože většina HTML dokumentů na internetu nedodržuje standardy. Běžný uživatel si chyb při zobrazení obsahu ve webovém prohlížeči nevšimne, protože moderní prohlížeče je automaticky odstraňují.

Vyhledání a následné získání určitých dat se nazývá obecně parsování. Jsou za tímto účelem vyvinuty i speciální knihovny. K parsování HTML slouží například knihovna *htmlparser* [7], který je použit v této bakalářské práci. Tento parser podporuje automatickou opravu webové stránky a poskytuje nástroje pro rychlé a snadné parsování.

Při parsování nastává problém, jak identifikovat potřebné informace. Každá stránka obsahuje velké množství tagů. Ty mohou mít až několik atributů, které by měli blíže specifikovat typ tagu a jeho informace. Dále mohou mít jako potomka další tag, což způsobuje to, že jsou do sebe většinou určitým způsobem zanořené. Proto je nutné najít způsob identifikace jednotlivých informací.

Prvním způsobem je projít posloupnost tagů a nalézt určitou informaci. Problémem tohoto způsobu je, že tato posloupnost se nemusí v daném HTML dokumentu vyskytovat pouze jednou. Z tohoto důvodu by se nemusela najít ta pravá informace.

Druhým způsobem je identifikace pomocí určitých atributů. Tento způsob je možný, pokud tag, který nese informaci, má nějaký atribut. Je zde možnost, že tagů se stejným atributem bude více jak jeden. Poté se potýkáme se stejným problémem jako u předchozího způsobu.

Největší pravděpodobnost nalezení informací, které chceme, je zkombinovat tyto dva způsoby. To znamená vyhledávat tagy určité posloupnosti s určitými atributy.

3.4 XML

XML je značkovací jazyk, který definuje soubor pravidel pro kódování dokumentů. Tento soubor pravidel je srozumitelný jak pro člověka, tak pro stroj. Je definován konsorciem W3C ve specifikaci XML 1.0 [8].

Slouží ke strukturalizaci dat. Mezi strukturovaná data patří například tabulky, adresáře, konfigurace, obchodní transakce, technické výkresy atd. XML je soubor pravidel tvorby textových formátů, které umožní data uspořádat ve strukturách. Není to programovací jazyk, a k jeho zvládnutí není třeba znalostí o programování. Usnadňuje počítači tvořit, číst a zapisovat data, a zajistit jednoznačnost struktury dat. XML se vyhnulo běžným nevýhodám popisných jazyků: je rozšiřitelné, nezávislé na platformě, a podporuje lokalizaci. Plně vyhovuje standardu Unicode.

XML se podobá HTML. Stejně jako HTML, i XML používá tzv. tagy (jména uzavřená mezi špičatými závorkami, např. <zamestnanec>) a atributy (ve tvaru jméno="hodnota"). Zatímco však HTML přesně specifikuje, co který tag či atribut znamená a jak bude v prohlížeči zobrazen text uvnitř, XML používá tagy pouze k ohraničení částí dat, a jejich interpretace je přenechána aplikaci, která data čte. Jinými slovy, pokud je v XML tag <b>, nepředpokládá se, že bude obsahovat tučné písmo. Podle situace to může znamenat např. bydliště, body, barva, nebo cokoliv jiného.

Jak HTML, tak XML jsou textové soubory. XML je však oproti HTML o něco přísnější, co se týká formátu. Zapomenutý tag nebo atribut bez uvozovek dělá XML soubor nevalidní, zatímco HTML to někde dokonce výslovně povoluje. Oficiální XML specifikace zakazuje aplikacím "domýšlet si", co tvůrce poškozeného XML souboru zamýšlel, a pokud objeví chybu, musí načítání zastavit a ohlásit chybu.

3.5 JAXB

Java i XML jsou technologie, které se často používají pro komunikaci mezi aplikacemi na různých operačních systémech. Proto je v Javě řada možností, jak s XML daty pracovat, např. SAX (*Simple API for XML*), DOM (*Document Object Model*) atd. Jednou z dalších technologií provazující Javu s XML je i JAXB (*Java Architecture for XML Binding*). JAXB nabízí metody pro konverzi XML dat na Java objekty a naopak, a umožňuje zápis a čtení XML z mnoha různých zdrojů, například ze souboru, streamu, nebo z URL (*Uniform Resource Locator*).

JAXB představuje zcela jiný přístup ke zpracování XML dokumentu, než jsou SAX a DOM. Jedná se o automatické mapování mezi XML dokumentem a odpovídajícími Java třídami vygenerovanými pomocí XSD (*XML Schema Definition*) schématu. Struktura dokumentu je již popsána tímto schématem.

JAXB verze 2.0 je kompatibilní s JDK 5 a je součástí JDK 6. Aktuální verze je 2.2 a je obsažena v JDK 7. Všechno co uživatel potřebuje pro práci s JAXB se nachází v balíku `javax.xml.bind`.

Nevýhodou tohoto přístupu je, že vytváří celý XML strom v paměti. Z tohoto důvodu, jsou aplikace s velkými výstupními daty velmi paměťově náročné.

3.6 StAX

StAX (*Streaming Api for Xml*) je další technologií pro zpracování XML. StAX oproti JAXB představuje proudové zpracování dokumentu, jak čtení, tak zápis. Tím pádem není tato metoda tolik náročná na paměť.

4 Implementace

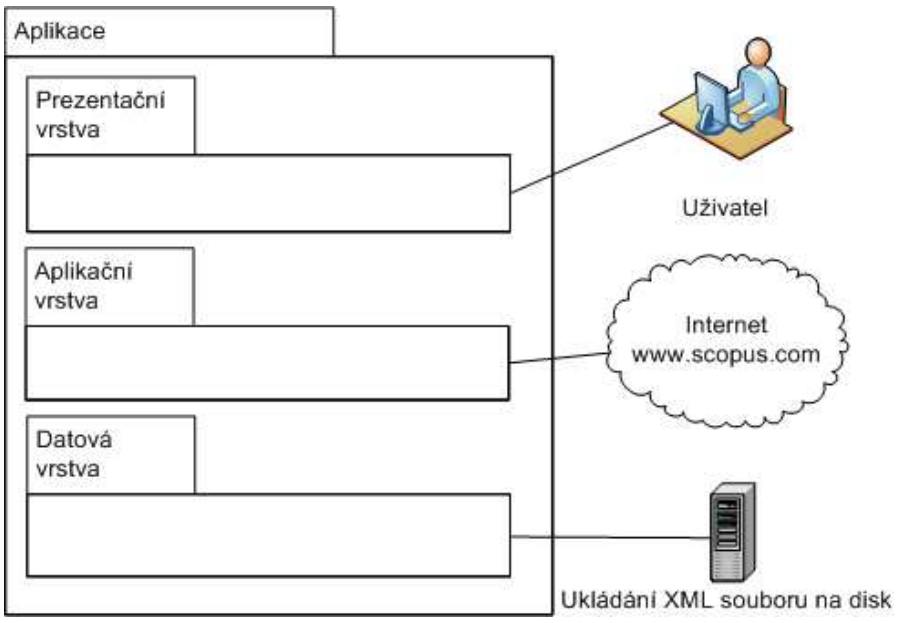
4.1 Použité technologie

Aplikace byla vytvořena v programovacím jazyce Java verze 1.6.0. Vyvinuta byla na platformě Windows 7 Professional ve vývojovém prostředí Eclipse Indigo verze 3.7.

Parsování se provádí přímo z HTML kódu stránek pomocí knihovny *htmlparser*. Tyto stránky se získávají pomocí knihovny HttpClient [9] od společnosti Apache.

4.2 Architektura

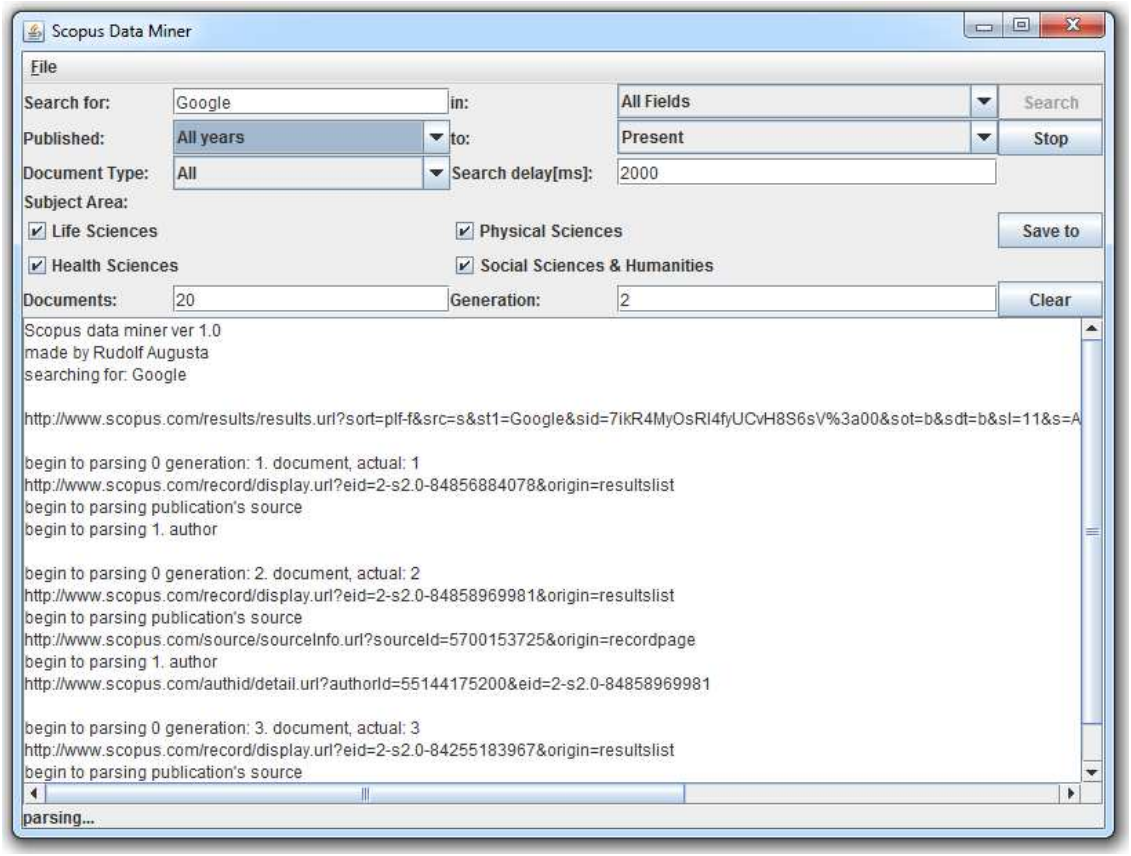
Aplikace má tří vrstvou architekturu (viz Obr. 4.1). Obsahuje prezentační, aplikační a datovou vrstvu.



Obr. 4.1: Pohled na architekturu aplikace vzhledem k okolí.

4.2.1 Prezentační vrstva

Předmětem této vrstvy je uživatelské rozhraní. Protože zájmem této aplikaci je získávání informací o dokumentech, inspirací pro GUI (*Graphic User Interface*) (viz. Obr. 4.2) je formulář *Document search* (viz. Obr. 2.1). V horní části lze nastavit omezení. Tuto část vytváří třída `GuiHead.java`. Uprostřed GUI je konzole, do které se vypisuje, co se právě zpracovává. V konzoli se uchovává pouze posledních sto řádků. Poslední částí GUI je stavový řádek.



Obr. 4.2: GUI aplikace.

4.2.2 Aplikační vrstva

V této vrstvě jsou dvě třídy: `DocumentParser.java`, `UrlLinkedList.java`. První z těchto tříd je zodpovědná za vyhledávání dokumentů. Druhá třída slouží k uchování seznamu URL adres, které jsou použity k průchodu mezi generacemi a k vyhledání jednotlivých dokumentů.

4.2.3 Datová vrstva

Tato vrstva obsahuje tři třídy, použité k získání a uchování informací: `Document.java`, `Author.java`, `Source.java`. Dále obsahuje třídu `XmlFileStax.java`, která slouží k ukládání informací, o dokumentech přijatých z aplikační vrstvy, do XML souboru.

4.3 Získávání informací

Při získávání informací je zapotřebí postupovat, jako při vyhledávání, ve službě Scopus, ve webovém prohlížeči na internetu. To obstarává aplikační vrstva, konkrétně třída `DocumentParser.java`. Instance této třídy se vytvoří jako nové vlákno.

4.3.1 Formulář

Napřed musím na server poslat požadavek z vyplněného formuláře. K tomu jsem použil metodu POST z knihovny HttpClient. Tato metoda sestaví požadavek z vloženého seznamu informací.

Na základě požadavku server odpoví. Odpověď obsahuje status kód, podle kterého zjistím, zda server požadavek přijal nebo ne. K pokračování potřebuji číslo kódu 202, to znamená přesměrování.

Dále si musím z odpovědi od serveru ještě získat adresu, na kterou mi server přesměroval. HTML kód této stránky získám metodou GET opět z knihovny HttpClient.

4.3.2 Vyhledané dokumenty

Po získání HTML kódu stránky s vyhledanými dokumenty mohu začít získávat informace. K tomu slouží metoda `fillDocument()`.

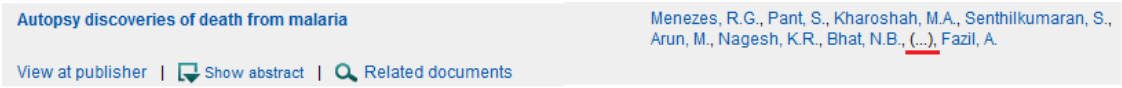
Jak jsem již psal (viz kapitola 2.1.2), na této stránce jsou názvy dokumentů, autoři, rok vydání, zdroj, ve kterém byl dokument vydán, a počet, kolikrát byl dokument citován.

Název dokumentu je odkaz, který v HTML značce obsahuje i URL adresu na detail dokumentu. Z této značky získávám název a URL. Z URL získávám elektronické ID.

Dále z této stránky získávám rok vydání dokumentu.

Nakonec získávám počet citací. Pokud je roven nule tak je to pouze textová značka, jinak je to odkaz na seznam dokumentů, které daný dokument citují. Protože tento odkaz potřebuji k získání elektronických ID citujících dokumentů a k získání další generace, tak si ho také uchovávám.

V případě informace o autorech nastává problém. Je zde uvedeno pouze jméno a jejich ID. Není uvedena afilace. Dále v případě, že dokument má více autorů s dlouhými jmény, někteří autoři nejsou uvedení a jsou nahrazení třemi tečkami (viz. Obr. 4.3).



Obr. 4.3: Nahrazení autorů tečkami u dokumentu.

Problém nastává i u zdroje dokumentu, který má popisky se stránkami, nebo článku, ve kterém byl dokument vydán, ve zkratkovité formě (viz. Obr. 4.4). Z těchto důvodů musím přejít na detail dokumentu.

Obr. 4.4: Zkratkovitá forma zápisu informací týkající se zdroje.

4.3.3 Detail dokumentu

Na této stránce jsou již autoři vyjmenováni všichni. Dále zde bývá také napsána jejich afilace, kterou měli, když dílo psaly. Afilace je k autorům přiřazena pomocí indexů. Autor může mít těchto indexů víc. To znamená, že autor v době psaní změnil afilaci. Nebo také nemusí mít žádný index, pak ke všem autorům je přiřazena jediná, která na stránce je. Pokud na této stránce afilace není, autorům se přiřadí afilace, kterou mají v detailu autora.

Z této stránky získávám také informace o zdroji: název, stránky, na kterých bylo dílo uvedeno ve zdroji atd. V požadavku zadavatele bylo získat také vydavatele zdroje. Pro tuto informaci musím přejít na stránku s detailem o zdroji, pokud tato stránka existuje.

4.3.4 Detail autora

Z této stránky získávám afilaci autora, pokud není uvedena v detailu dokumentu. Jelikož autorů je často více jak jeden, musím projít tuto stránku u všech autorů. To má za následek zpomalení programu.

4.3.5 Detail zdroje

V případě, že je daný zdroj evidován v databázi Scopus, lze se dostat k jeho informacím na speciální stránce. Na této stránce získávám vydavatele zdroje.

4.3.6 Dokumenty citující daný dokument

Na tuto stránku se dostanu pouze v případě, že dokument byl alespoň jednou citován. To znamená, že počet citací je odkaz na další stránku. Ta má stejnou strukturu jako první stránka s vyhledanými dokumenty.

Protože jsem si navrhnul strukturu XML souboru tak, že u každého dokumentu bude seznam elektronických ID citujících dokument. Získávám z těchto stránek elektronické ID ze značek odkazů a ukládám si je do seznamu uchovávaném u daného citovaného dokumentu.

4.4 Získání další generace

K tomu abych získal další generaci publikací, používám odkaz na citující publikace, jak jsem již psal (viz kapitola 4.3.2). Pomocí metody GET získávám HTML stránku stejné struktury jako je první stránka s vyhledanými publikacemi. Proto nad ní zavolám metodu `fillDocuments()` na získání informací.

4.5 Ukládání výsledků

Získaná data se uloží do pomocné třídy `Document.java`. Instance této třída se po získání všech potřebných informací pošle do třídy `XmlFileStax.java`, Kde se pomocí StAXu postupně ukládají všechny přijaté dokumenty. Toto postupné ukládání slouží jako záloha v případě, kdyby aplikace byla neočekávaně ukončena.

4.5.1 Formát ukládání

Nalezené informace se ukládají do souboru ve formátu XML (viz. Obr 4.5).

```
<?xml version="1.0" encoding="UTF-8"?>
<publications>
    <document eid="" generation="" documentNumber="">
        <title year="">název dokumentu</title>
        <sourceInfo publisher="" volume="" issue="" articleNumber="" pages="">název zdroje</sourceInfo>
        <authors count="">
            <author id="" affiliation="">jméno autora</author>
        </authors>
        <affiliations count="">
            <affiliation key="">afilace</affiliation>
        </affiliations>
        <citations count="">
            <citedBy>elektronické id</citedBy>
        </citations>
    </document>
</publications>
```

Obr. 4.5: Formát výstupního XML souboru.

Z příkladu je vidět, že kořenovým elementem XML souboru je <`publications`>, který nemá žádné atributy. Tento element dále může obsahovat nula až N elementů <`document`>.

Element <`document`> již obsahuje informace o jednotlivých dokumentech, které se získali při vyhledávání pomocí aplikace. Samotný element má tři atributy: `eid`, `generation`, `documentNumber`. První atribut slouží k jednoznačné identifikaci dokumentu v databázi Scopus. Následující atribut určuje, v kolikáté generaci od počátku vyhledávání se dokument nalezl. Poslední atribut je pořadové číslo, kolikátý v pořadí od počátku vyhledávání, byl dokument nalezen.

Informace, které element <document> obsahuje, jsou uvedeny do pěti podelementů: <title>, <sourceInfo>, <authors>, <affiliations> a <citations>. V následující části tyto elementy podrobněji popíši.

Prvním uvedeným elementem je <title>, který obsahuje název dokumentu. Tento element má jediný atribut. Tímto atributem je rok, ve kterém byl dokument vydán.

Následujícím elementem je <sourceInfo>. V tomto elementu je obsažen název zdroje. Tím je například konference, na které byl dokument vydán, název časopisu, ve kterém byl dokument vydán atd. Tento element má pět atributů:

- publisher - jedná se o vydavatele, který je zodpovědný za vydání daného zdroje.
- volume - jedná se o svazek daného zdroje, ve kterém byl dokument vydán.
- issue - číslo vydání daného zdroje.
- articleNumber - pořadové číslo článku ve zdroji.
- pages - označuje stránky zdroje, na kterých se nachází vyhledaný dokument.

Dalším elementem je <authors>, který má pouze jeden atribut: count. Tento atribut vyčíslí, kolik má daný dokument autorů. Samotný element obsahuje další podelementy s názvem <author>.

Element <author> má dva atributy: id a affiliation. První z atributů jednoznačně identifikuje autora v databázi Scopus. Druhý je již zmíněný index (písmeno nebo posloupnost písmen viz Obr. 2.2), který přiřazuje k autorovy určitou afilaci. Element obsahuje jako svojí hodnotu jméno autora. Pokud u dokumentu nejsou uvedeni autoři, vypíše se autor se jménem: „no author names available.“

Následuje element <affiliations>, který má stejně jako element <authors> jediný atribut: count. V tomto případě představuje atribut počet afiliací, z kterých autoři pocházeli, v době psaní daného dokumentu. Tento element obsahuje také podelementy, ale tyto mají název <affiliation>.

Element <affiliation> má atribut key. Ten představuje daný index, který se použije pro přiřazení autora k dané afilaci. Element obsahuje jako hodnotu název afilace.

Posledním podelementem, který je obsažen v elementu <document>, je <citations>. Jeho atributem je count a představuje počet, kolikrát byla daná publikace citována. Také má podelementy s názvem <citedBy>.

Element <citedBy> nemá žádný atribut, obsahuje pouze hodnotu. Touto hodnotou je eid dokumentů, které daný dokument citují.

5 Testování

Jedním z úkolů této práce je otestování implementované aplikace. Aplikace byla testována během celé doby vývoje, neboť implementace probíhala iterativním způsobem a každá nová funkcionalita v příslušné iteraci byla následně testována. Výsledky testů byly při schůzce se zadavatelem předvedeny, čímž docházelo i ke korekci implementace.

Bylo prováděno ruční testování. Při výskytu jakékoliv chyby, byla chyba odstraněna. Následně byly provedeny další testy, aby odhalili, jestli nevznikli nějaké další chyby.

Jedna z chyb, odhalená při testování, byla získávání špatných informací. Tato chyba nastala, protože provozovatelé aplikace Scopus změnili strukturu stránek.

Ukázka testovacího výstupu (viz příloha B) na dotaz: „Automatically building research reading lists“. Řetězec byl vyhledán jako název dokumentu (Article Title). Dále byl omezený, aby se vyhledaly dokumenty pouze do druhé generace.

6 Závěr

V této části bych chtěl zhodnotit dosažené cíle bakalářské práce s ohledem na zadání. Aplikace byla průběžně předváděna a konzultována se zadavatelem, což vedlo k úspěšnému splnění jednotlivých bodů zadání.

V počáteční fázi jsem se seznámil s webovou aplikací Scopus jako běžný uživatel, protože jsem potřeboval zjistit, jaké informace lze z aplikace získat. Také jsem musel prozkoumat zdrojové kódy těchto stránek, kvůli přesnější identifikaci informací.

Následovala fáze analýzy, jakými způsoby lze informace z aplikace získat. Dále jak lze tyto informace uložit. S tímto je spojen i návrh formátu XML dokumentu.

Následně jsem měl implementovat získané informace a vytvořit tak aplikaci, která získá vyhledaná data z webové aplikace Scopus a uloží je do XML souboru, který bude mít navrženou strukturu. Myslím si, že jsem tento cíl splnil. Program dokáže posílat na server dotazy a přijímat odpovědi. Z těchto odpovědí získat požadovaná data a ty uložit do XML souboru, který má navrženou strukturu.

Autorský přínos práce lze pozorovat především v rozšíření znalostí a získání rozsáhlých praktických zkušeností v problematice webových aplikací a technologií pro jejich implementaci. Ty se v budoucnu mohou stát velkou výhodou při práci v tomto odvětví.

Jedním z možných vylepšení této práce, bych navrhoval například dodělat do aplikace ostatní možnosti vyhledávání, které webová aplikace Scopus nabízí. Dále bych navrhnul předělat uchovávání nutných informací o zpracovaných dokumentech. Těmito informacemi jsou elektronické id, již zpracovaných dokumentů, a URL adresa na další generace. Z důvodu paměťové náročnosti, bych tyto informace ukládal do souboru na disk. Také by se tím umožnilo opětovné spuštění aplikace při jejím přerušení.

Přehled zkratek

XML	*Extensible Markup Language*
HTTP	*HyperText Transfer Protocol*
HTML	*HyperText Markup Language*
URI	*Uniform Resource Identifier*
WWW	*World Wide Web*
EID	*elektronický identifikátor*
HTTPS	*Hypertext Transfer Protocol Secure*
SGML	*Standard Generalized Markup Language*
CERN	*Centre Européenne pour la Recherche Nucléaire, Evropské centrum jaderného výzkumu*
DTD	*Document Type Definition*
SAX	*Simple API for XML*
DOM	*Document Object Model*
JAXB	*Java Architecture for XML Binding*
URL	*Uniform Resource Locator*
XSD	*XML Schema Definition*
StAX	*Streaming Api for Xml*
URL	*Uniform Resource Locator*
GUI	*Graphic User Interface*

Literatura

[1] *Scopus* [online]. 2012, [cit. 2012-04-11]. <http://www.scopus.com/>

[2] *Elsevier* [online]. 2012, [cit. 2012-04-11]. <http://www.elsevier.com/>

[3] GOURLEY, D. - TOTTY, B. - SAYER, M. - AGGARWAL, A. - REDDY, S. *HTTP: The Definitive Guide.* Sebastopol: O'Reilly Media, 2002. 658 s. ISBN: 978-1-56592-509-0.

[4] KENNEDY, B. – MUSCIANO, C. *HTML & XHTML: The Definitive Guide*. 6th Edition. Sebastopol: O'Reilly Media, 2006. 678 s. ISBN 0-596-52732-2.

[5] *SGML* [online]. 2004, poslední úpravy 26.03.2004, [cit. 2012-05-01]. <http://www.w3.org/MarkUp/SGML/>

[6] *HTML tags* [online]. 2008, [cit. 2012-05-01]. <http://www.w3.org/History/19921103-hypertext/hypertext/www/MarkUp/Tags.html>

[7] *htmlparser* [online]. 2006, poslední úpravy 17.09.2006 [cit. 2012-05-01]. <http://htmlparser.sourceforge.net/>

[8] *XML 1.0 Specification* [online]. 2008, poslední úpravy 28.11.2008 [cit. 2012-05-01]. <http://www.w3.org/TR/REC-xml/>

[9] *HttpClient* [online]. 2012, poslední úpravy 22. 2. 2012 [cit. 2012-04-12]. <http://hc.apache.org/index.html>

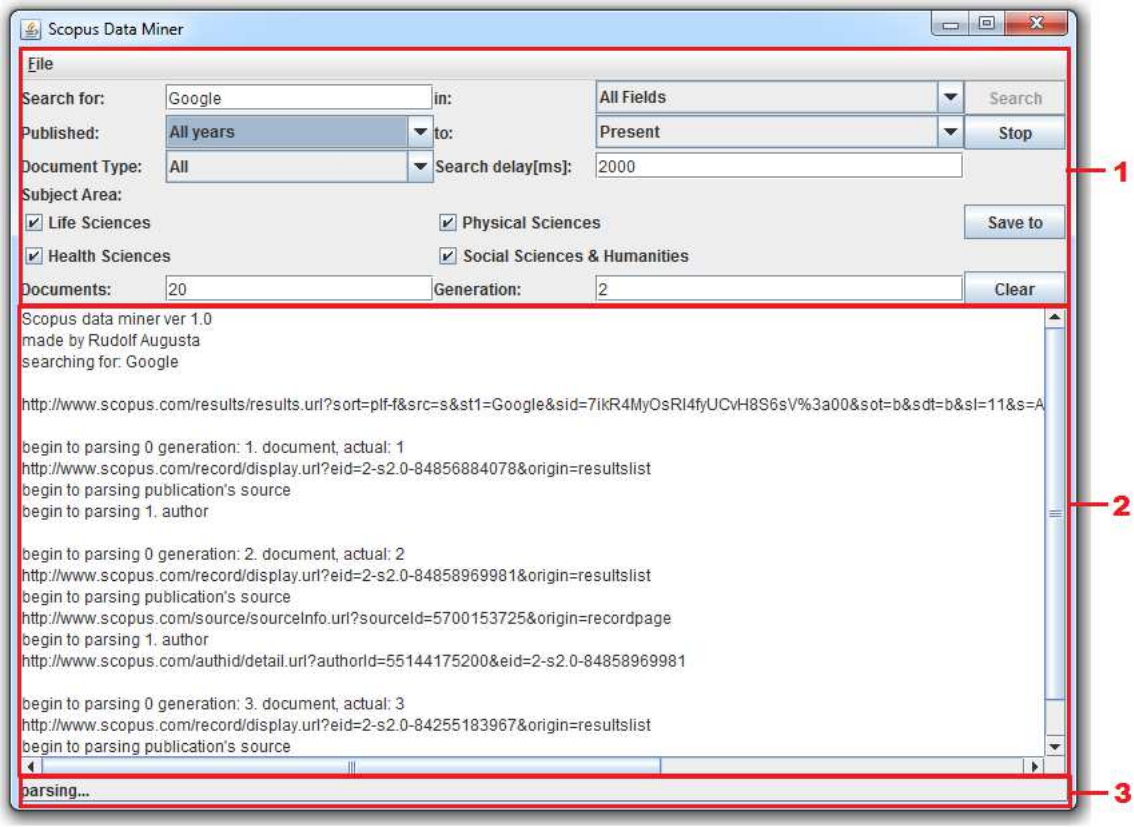
Příloha A: Uživatelský manuál

A.1 Spuštění programu

Aplikace klienta se spustí přes příkazový řádek příkazem: `java –jar Augustar_prj5_A09B0169P_ScopusDataMiner.jar` nebo dvojklikem na soubor ScopusDataMiner.jar v případě předem nastaveného systému. Je nutné mít nainstalováno Java Runtime 6.0. Aplikace byla zkoušena na platformě Windows 7. Po spuštění Vás uvítá okno aplikace, jak je ilustrováno na následujícím obrázku (Obr. A.1).

A.2 Popis GUI

Grafické uživatelské prostředí je rozděleno na tři časti (viz Obr. A.1).



Obr A.1: GUI vytvořené aplikace: (1)Nastavení hledání a omezení, (2)Konzole, (3)Stavový řádek.

A.2.1 Nastavení hledání a omezení

V horní časti je nastavení, co chce uživatel hledat: *Search for*. Dále lze nastavit, v jaké části se hledaný řetězec má vyskytovat: *in*, jestli řetězec hledáme v abstraktu, názvu dokumentu, nebo hledaný řetězec je jméno autora, nebo název konference, na které byl dokument vydán atd. Také lze nastavit *Published*, to slouží k omezení, aby se zobrazili dokumenty z určitého období. V neposlední řadě můžeme nastavit pomocí *Document type*, jakého typu hledané dokumenty mají být. Také si lze z GUI nastavit prodleva mezi vyhledávání, aby nebyl server tak často dotazován. Dále lze nastavit, v jakém odvětví výzkumu chceme hledat. Poslední možná nastavení jsou: šířka nulté generace a kolik generací se má prohledat.

A.2.2 Konzole

V této části se nachází konzole, na kterou se vypisují informace, co se v daném okamžiku zpracovává. Také se zde vypisují odkazy stránek, které v danou chvíli aplikace zpracovává. V konzoli se uchovává pouze posledních sto řádku.

A.2.3 Stavový řádek

Ve spodní části se nachází stavový řádek. Po spuštění aplikace je zde napsáno „Welcome in scopus data miner“. Jakmile se začne vyhledávat tak se zde zobrazí „parsing“ a za tímto slovem se pořád „běhají“ tři tečky, aby bylo vidět, že program pracuje. V případě nějaké chyby se zde objeví chybové hlášení.

Příloha B: Výstupní XML

Následující text je výstupní XML soubor získaný po vyhledání aplikací na dotaz: „Automatically building research reading lists“. Řetězec byl vyhledán jako název dokumentu (Article Title). Dále byl omezený, aby se vyhledaly dokumenty pouze do druhé generace.

```
<?xml version="1.0" encoding="UTF-8"?>
<publications>
	<document eid="2-s2.0-78649953561" generation="0" documentNumber="1">
		<title year="2010">Automatically building research reading lists</title>
		<sourceInfo publisher="" volume="" issue="" articleNumber  pages="159-
		166">RecSys'10 - Proceedings of the 4th ACM Conference on Recommender
		Systems</sourceInfo>
		<authors count="6">
			<author id="35302511500" affiliation="a">Ekstrand, M.D.</author>
			<author id="36666641700" affiliation="a">Kannan, P.</author>
			<author id="23398620500" affiliation="b">Stemper, J.A.</author>
			<author id="12776006700" affiliation="b">Butler, J.T.</author>
			<author id="7003985614" affiliation="a">Konstan, J.A.</author>
			<author id="7005582593" affiliation="a">Riedl, J.T.</author>
		</authors>
		<affiliations count="2">
			<affiliation key="b ">University of Minnesota Libraries, United
			States</affiliation>
			<affiliation key="a ">GroupLens Research, Department of Computer
			Science and Engineering, University of Minnesota, United
			States</affiliation>
		</affiliations>
		<citations count="8">
			<citedBy>2-s2.0-84858702721</citedBy>
			<citedBy>2-s2.0-83055161602</citedBy>
			<citedBy>2-s2.0-80455149922</citedBy>
			<citedBy>2-s2.0-80054946579</citedBy>
			<citedBy>2-s2.0-80052711853</citedBy>
			<citedBy>2-s2.0-80052086298</citedBy>
			<citedBy>2-s2.0-79955136318</citedBy>
			<citedBy>2-s2.0-80052405657</citedBy>
		</citations>
	</document>
	<document eid="2-s2.0-84858702721" generation="1" documentNumber="2">
		<title year="2012">Recommender systems: From algorithms to user
		experience</title>
		<sourceInfo publisher="Kluwer Academic Publishers" volume="22" issue="1-
		2" articleNumber="" pages="101-123">User Modelling and User-Adapted
		Interaction</sourceInfo>
		<authors count="2">
			<author id="55061427400" affiliation="a">Konstan, J.A.</author>
			<author id="7005582593" affiliation="a">Riedl, J.</author>
		</authors>
		<affiliations count="1">
			<affiliation key="a">GroupLens Research, Department of Computer
			Science and Engineering, University of Minnesota, Minneapolis, MN
			55455, United States</affiliation>
		</affiliations>
		<citations count="3">
			<citedBy>2-s2.0-84858701909</citedBy>
			<citedBy>2-s2.0-84858705793</citedBy>
			<citedBy>2-s2.0-84858700483</citedBy>
		</citations>
	</document>
	<document eid="2-s2.0-83055161602" generation="1" documentNumber="3">
		<title year="2011">Adding structure to top-k: From items to
		expansions</title>
		<sourceInfo publisher="Association for Computing Machinary, Inc."
		volume="" issue="" articleNumber="" pages="1699-1708">International
		Conference on Information and Knowledge Management,
		Proceedings</sourceInfo>
		<authors count="3">
			<author id="54581126900" affiliation="a">Liang, X.</author>
			<author id="36668014800" affiliation="a">Xie, M.</author>
			<author id="7003716659" affiliation="a">Lakshmanan,
			L.V.S.</author>
		</authors>
```

```
        <affiliations count="1">
                <affiliation key="a">Dept. of Computer Science, University of
                British Columbia, Vancouver, BC, Canada</affiliation>
        </affiliations>
        <citations count="0"/>
</document>
<document eid="2-s2.0-80455149922" generation="1" documentNumber="4">
        <title year="2011">A multicriteria recommendation method for data with
        missing rating scores</title>
        <sourceInfo publisher="" volume="" issue="" articleNumber="6053931"
        pages="60-67">Proceedings - 2011 International Conference on Data and
        Knowledge Engineering, ICDKE 2011</sourceInfo>
        <authors count="1">
                <author id="7005461524" affiliation="a">Takasu, A.</author>
        </authors>
        <affiliations count="1">
                <affiliation key="a">National Institute of Informatics, Tokyo,
                Japan</affiliation>
        </affiliations>
        <citations count="0"/>
</document>
<document eid="2-s2.0-80054946579" generation="1" documentNumber="5">
        <title year="2011">Related paper recommendation to support online-
        browsing of research papers</title>
        <sourceInfo publisher="" volume="" issue="" articleNumber="6041413"
        pages="130-136">4th International Conference on the Applications of
        Digital Information and Web Technologies, ICADIWT 2011</sourceInfo>
        <authors count="3">
                <author id="36091923800" affiliation="a">Ohta, M.</author>
                <author id="35145607400" affiliation="a">Hachiki, T.</author>
                <author id="7005461524" affiliation="b">Takasu, A.</author>
        </authors>
        <affiliations count="2">
                <affiliation key="b ">National Institute of Informatics, 101-8430
                Tokyo, Japan</affiliation>
                <affiliation key="a ">Graduate School of Natural Science and
                Technology, Okayama University, Okayama 700-8530,
                Japan</affiliation>
        </affiliations>
        <citations count="0"/>
</document>
<document eid="2-s2.0-80052711853" generation="1" documentNumber="6">
        <title year="2011">Finding relevant papers based on citation
        relations</title>
        <sourceInfo publisher="Springer Verlag" volume="6897 LNCS" issue=""
        articleNumber="" pages="403-414">Lecture Notes in Computer Science
        (including subseries Lecture Notes in Artificial Intelligence and Lecture
        Notes in Bioinformatics)</sourceInfo>
        <authors count="3">
                <author id="50262209100" affiliation="a">Liang, Y.</author>
                <author id="36067406200" affiliation="a">Li, Q.</author>
                <author id="17135487000" affiliation="bc">Qian, T.</author>
        </authors>
        <affiliations count="3">
                <affiliation key="b ">State Key Laboratory of Software
                Engineering, Wuhan University, Wuhan, China</affiliation>
                <affiliation key="a ">Department of Computer Science, City
                University of Hong Kong, Hong Kong, Hong Kong</affiliation>
                <affiliation key="c ">State Key Laboratory for Novel Software
                Technology, Nanjing University, Nanjing, China</affiliation>
        </affiliations>
        <citations count="0"/>
</document>
<document eid="2-s2.0-80052086298" generation="1" documentNumber="7">
        <title year="2011">User-centered profile representation for
        recommendations across multiple content domains</title>
        <sourceInfo publisher="" volume="15" issue="1" articleNumber="" pages="1-
        14">International Journal of Knowledge-Based and Intelligent Engineering
        Systems</sourceInfo>
        <authors count="2">
                <author id="7101987109" affiliation="a">Fukazawa, Y.</author>
                <author id="7006719514" affiliation="b">Ota, J.</author>
        </authors>
        <affiliations count="2">
                <affiliation key="b ">University of Tokyo, 5-1-5 Kashiwanoha,
                Kashiwa, Chiba, Japan</affiliation>
```

```
                <affiliation key="a ">NTT DOCOMO, Inc., 3-6 Hikari-no-oka,
                Yokosuka, Kanagawa, Japan</affiliation>
        </affiliations>
        <citations count="0"/>
</document>
<document eid="2-s2.0-79955136318" generation="1" documentNumber="8">
        <title year="2011">OSUSUME: Cross-lingual recommender system for research
        papers</title>
        <sourceInfo publisher="" volume="" issue="" articleNumber="" pages="39-
        42">ACM International Conference Proceeding Series</sourceInfo>
        <authors count="4">
                <author id="37105024500" affiliation="a">Uchiyama, K.</author>
                <author id="8892529300" affiliation="b">Nanba, H.</author>
                <author id="6701312731" affiliation="a">Aizawa, A.</author>
                <author id="37104976000" affiliation="c">Sagara, T.</author>
        </authors>
        <affiliations count="3">
                <affiliation key="b ">Hiroshima City University, 3-4-1 Ozuka-
                higashi, Asaminami-ku, Hiroshima, Japan</affiliation>
                <affiliation key="a ">National Institute of Informatics, 2-1-2
                Hitotsubashi, Chiyoda-ku, Tokyo, Japan</affiliation>
                <affiliation key="c ">Picolab, Co., Ltd., 1-4-6 Kita-Aoyama,
                Minato-ku, Tokyo, Japan</affiliation>
        </affiliations>
        <citations count="0"/>
</document>
<document eid="2-s2.0-80052405657" generation="1" documentNumber="9">
        <title year="2010">Collaborative filtering recommender systems</title>
        <sourceInfo publisher="Now Publishers Inc." volume="4" issue="2"
        articleNumber="" pages="81-173">Foundations and Trends in Human-Computer
        Interaction</sourceInfo>
        <authors count="3">
                <author id="35302511500" affiliation="a">Ekstrand, M.D.</author>
                <author id="7005582593" affiliation="a">Riedl, J.T.</author>
                <author id="7003985614" affiliation="a">Konstan, J.A.</author>
        </authors>
        <affiliations count="1">
                <affiliation key="a">University of Minnesota, 4-192 Keller Hall,
                200 Union St., Minneapolis, MN 55455, United States</affiliation>
        </affiliations>
        <citations count="0"/>
</document>
<document eid="2-s2.0-84858701909" generation="2" documentNumber="10">
        <title year="2012">Critiquing-based recommenders: Survey and emerging
        trends</title>
        <sourceInfo publisher="Kluwer Academic Publishers" volume="22" issue="1-
        2" articleNumber="" pages="125-150">User Modelling and User-Adapted
        Interaction</sourceInfo>
        <authors count="2">
                <author id="14041395200" affiliation="a">Chen, L.</author>
                <author id="7005276079" affiliation="b">Pu, P.</author>
        </authors>
        <affiliations count="2">
                <affiliation key="b ">Human Computer Interaction Group, School of
                Computer and Communication Sciences, Swiss Federal Institute of
                Technology in Lausanne (EPFL), Lausanne 1015,
                Switzerland</affiliation>
                <affiliation key="a ">Department of Computer Science, Hong Kong
                Baptist University, Hong Kong, Hong Kong</affiliation>
        </affiliations>
        <citations count="1">
                <citedBy>2-s2.0-84858702721</citedBy>
        </citations>
</document>
<document eid="2-s2.0-84858705793" generation="2" documentNumber="11">
        <title year="2012">Discovery of Web user communities and their role in
        personalization</title>
        <sourceInfo publisher="Kluwer Academic Publishers" volume="22" issue="1-
        2" articleNumber="" pages="151-175">User Modelling and User-Adapted
        Interaction</sourceInfo>
        <authors count="1">
                <author id="55061299700" affiliation="a">Paliouras, G.</author>
        </authors>
        <affiliations count="1">
                <affiliation key="a">Institute of Informatics and
                Telecommunications, National Centre for Scientific Research
```

```
                        Demokritos, Patr. Grigoriou and Neapoleos str., Ag. Paraskevi,
                        Attiki 15310, Greece</affiliation>
                </affiliations>
                <citations count="3">
                        <citedBy>2-s2.0-84858698895</citedBy>
                        <citedBy>2-s2.0-84858700483</citedBy>
                        <citedBy>2-s2.0-84858701041</citedBy>
                </citations>
        </document>
        <document eid="2-s2.0-84858700483" generation="2" documentNumber="12">
                <title year="2012">Personalization in cultural heritage: The road
                travelled and the one ahead</title>
                <sourceInfo publisher="Kluwer Academic Publishers" volume="22" issue="1-
                2" articleNumber="" pages="73-99">User Modelling and User-Adapted
                Interaction</sourceInfo>
                <authors count="3">
                        <author id="6603677372" affiliation="a">Ardissono, L.</author>
                        <author id="6602259371" affiliation="b">Kuflik, T.</author>
                        <author id="9636756200" affiliation="c">Petrelli, D.</author>
                </authors>
                <affiliations count="3">
                        <affiliation key="b ">University of Haifa, Haifa,
                        Israel</affiliation>
                        <affiliation key="a ">Universitŕ di Torino, Turin,
                        Italy</affiliation>
                        <affiliation key="c ">Sheffield Hallam University, Sheffield,
                        United Kingdom</affiliation>
                </affiliations>
                <citations count="1">
                        <citedBy>2-s2.0-84858705793</citedBy>
                </citations>
        </document>
</publications>
```