Slovenská technická univerzita v Bratislave
FAKULTA INFORMATIKY A INFORMAČNÝCH TECHNOLÓGIÍ

FIIT-5208-64391

Bc. Marek Lóderer

Vyhľadávanie skrytých vzťahov v digitálnych knižniciach

Diplomová práca

Študijný program: Informačné systémy

Študijný odbor: 9.2.6 Informačné systémy

Miesto vypracovania: Ústav informatiky a softvérového inžinierstva, FIIT STU, Bratislava; Cosmotron Slovakia, s r.o.

Vedúca projektu: Ing. Nadežda Andrejčíková, PhD.

máj 2014

Slovenská technická univerzita v Bratislave
FAKULTA INFORMATIKY A INFORMAČNÝCH TECHNOLÓGIÍ

Anotácia

Vyhľadávanie skrytých vzťahov v digitálnych knižniciach

Študijný program: Informačné systémy

Autor: Bc. Marek Lóderer

Vedúca diplomovej práce: Ing. Nadežda Andrejčíková, PhD.

máj 2014

Cieľom práce je vytvoriť automatizovaný systém na transformáciu bibliografických záznamom do VIVO ontológie a vytvoriť tak prepojenie s bázou dát Linked Data. Bibliografické dáta budú čerpané z akademických knižníc. Nová reprezentácia údajov umožní nové spôsoby vyhľadávania a sprístupní sémantické vzťahy, ktoré sa nachádzajú v bibliografických záznamoch, no nie sú bežne využívané. Práve tieto vzťahy umožnia odhaľovať súvislosti medzi výskumníkmi.

Systém sa bude primárne zameriavať na problém rozoznávania mien autorov, vydavateľov a miest publikovania. Na Slovensku nie je vytvorený súbor autorít, ktorý by ponúkal unifikované selekčné prvky na identifikáciu autorov a iných entít v bibliografických záznamoch. Cieľom je preto vytvoriť metódy, ktoré dokážu s čo najväčšou presnosťou určovať inštancie a ich rôzne pomenovania. Napríklad zjednotiť rôzne mená toho istého autora spôsobené preklepmi pri vytváraní bibliografických záznamov alebo rozlišovať dvoch a viacerých autorov s rovnakým menom.

Slovak University of Technology Bratislava
FACULTY OF INFORMATICS AND INFORMATION TECHNOLOGY

ANNOTATION

Discovering hidden relationships in digital libraries

Degree Course: Information Systems

Author: Bc. Marek Lóderer

Supervisor: Ing. Nadežda Andrejčíková, PhD.

May 2014

The aim of this work is to create an automated system to transform bibliographic record into VIVO ontology and to create a connection with the Linked Data. Bibliographic data will be obtained from academic libraries. The new data representation will enable new ways of searching and it will make available the semantic relationships that are found in bibliographic records, but are not commonly used. These relationships will allow to detect links among researchers.

The system will primarily focus on the problem of name resolution of authors, publishers and publishing sites. Slovakia does not have an authority file, which would offer selective unified elements to identify authors and other entities in the bibliographic records. The aim is to create methods for determining the instances and their various names. For example, uniting different names of the same author due to typos in bibliographic records or to distinguish two or more authors with the same name.

Slovenská technická univerzita v Bratislave
FAKULTA INFORMATIKY A INFORMAČNÝCH TECHNOLÓGIÍ

ČESTNÉ PREHLÁSENIE

Čestne prehlasujem, že na diplomovej práci som pracoval samostatne na základe vlastných teoretických a praktických poznatkov, konzultácii a štúdia odbornej literatúry, ktorej úplný prehľad je uvedený v zozname použitej literatúry.

Bratislava, 14. máj 2014 .....

Bc. Marek Lóderer

Poďakovanie

Touto cestou chcem poďakovať mojej odbornej vedúcej, Ing. Nadežde Andrejčíkovej, PhD., za jej odborné vedenie, trpezlivosť pri konzultáciách, cenné rady a usmernenia. Toto všetko výraznou mierou prispelo k skvalitneniu tejto práce aj výsledného softvérového produktu.

Obsah

Použité skratky:

AACR2	Anglo-American Cataloguing Rules, Second Edition
ANSI	American National Standards Institute
BNB	British National Bibliography
CERIF	Common European Research Information Format
CREPČ	Centrálny register evidencie publikačnej činnosti
CVTI SR	Centrum vedecko-technických informácií SR
DOM	Document Object Model
DNB	Deutsche Nationalbibliothek
FRBR	Functional Requirements for Bibliographic Records
IFLA	International Federation of Library Associations and Institutions
ISBD	International Standard Bibliographic Description
ISBN	International Standard Book Number
ISSN	International Standard Serial Number
JWD	Jaro-Winkler Distance
LCSH	Library of Congress Subject Headings
M2V	Marc to VIVO
MARC	MAchine Redable Cataloging
MARC21	MAchine Redable Cataloging for 21st Century
NISO	National Information Standards Organization
OCLC	Online Computer Library Center
OAI PMH	Open Archives Initiative Protocol for Metadata Harvesting
OPAC	On-line Public Accessible Catalogue
RDA	Resource Description and Access
SWRC	Semantic Web for Research Communities
URI	Uniform Resource Identifier
VaV	Veda a výskum
VIAF	Virtual International Authority File
XML	Extensible Markup Language

Motivácia

Vývoj technológií urýchľuje šírenie informácií a podporuje neustály pokrok v oblasti vedy a výskumu. Výskumníci sú nútení sledovať aktuálne výskumné trendy, dosahované výsledky a nové progresívne výskumné témy. K tomu je potrebné poznať výskumníkov a komunity, ktoré sa venujú danej oblasti výskumu. Jedným zo zdrojov týchto informácii sú digitálne knižnice a bibliografické databázy. Vhodným spracovaním bibliografických dát je možné získať potrebné informácie a objaviť vzťahy medzi výskumnými pracoviskami, medzi samotnými výskumníkmi, ale aj súvislosti medzi výskumnými témami.

Digitálne knižnice obsahujú veľké množstvo kvalitných a aktuálnych informácií, týkajúcich sa výskumnej činnosti. Vývoj nových technológii v oblasti digitálnych knižníc neustále napreduje, vznikajú nové služby pre používateľov knižníc, vytvárajú sa nové formáty na efektívne uchovávanie a výmenu dát, podporuje sa digitalizácia dokumentov a uľahčuje sa publikačná činnosť. Toto všetko spôsobuje nárast objemu údajov v digitálnych knižniciach. Vďaka interoperabilite a vzájomnému prepojeniu môžu knižnice navzájom zdieľať svoj obsah a vytvárať tak širokú sieť informácii.

Vyhľadávacie stroje v knižniciach sú orientované prevažne na porovnávanie textových reťazcov, čo nemusí prinášať vždy relevantné informácie, pretože stroje nemusia rozumieť skrytým vzťahom a súvislostiam. Tu sa naskytuje možnosť zlepšiť možnosti reprezentácie a vyhľadávania údajov pomocou princípov sémantického webu. Sémantický web pracuje s údajmi, ktoré dokáže medzi sebou prepájať pomocou vzťahov. Na základe týchto vzťahov vzniká sieť údajov, ktorá poskytuje nové možnosti vyhľadávania informácií. Knižnice obsahujú veľké množstvo dát a metadát, ktoré môžu poslúžiť na vytvorenie sémantických vzťahov a naplnenie vhodnej ontológie.

Jedným z najčastejšie využívaných formátov na spracovanie metadát pre potreby sémantického webu je formát RDF (Resource Description Framework). Tento formát zapisuje údaje vo forme tripletov (podmet-prísudok-predmet). Formát RDF sa stáva čoraz populárnejší aj v oblasti digitálnych knižníc.

Cieľom práce je oboznámiť sa s možnosťami sémantizácie údajov v digitálnych knižniciach a vytvoriť metódy automatickej sémantizácie a odhaľovaniu skrytých vzťahov, ktoré budú vyžadovať minimálny zásah ľudského faktora.

1 Analýza

V časti analýza sa nachádzajú informácie o súčasných možnostiach webu a digitálnych knižniciach, o údajoch, ktoré v nich môžeme nájsť. V ďalšej časti sa nachádzajú informácie o ontológiách, ktoré sú určené pre oblasť digitálnych knižníc a vedy a výskumu. V tejto kapitole sa nachádzajú informácie o technológiách využívaných v ontológiách a oblasti sémantického webu. V závere analýzy sa nachádzajú informácie o procese sémantizácie bibliografických údajov, problémoch, ktoré pri tom vznikajú a projektoch, ktoré sa venujú tejto problematike.

1.1 Informácie o vede a výskume

Súčasný web obsahuje veľké množstvo odborných informácií z oblasti vedy a výskumu (VaV), ku ktorým môžu používatelia jednoducho pristupovať. Vzniklo veľké množstvo webových sídiel, bibliografických a citačných databáz, ktoré združujú informácie o výskumníkoch, ich činnosti a výsledkoch z oblasti VaV. Pokrok v technológiách spôsobil nárast informácií, ktoré sú dostupné na webe. Tento nárast je pozitívnym javom, pretože ponúka používateľom väčšiu bázu dát, no na druhej strane sa web stáva neprehľadný a spôsobuje problémy pri vyhľadávaní relevantných informácií, najmä, keď sa tieto informácie nenachádzajú na jednom mieste a sú medzi nimi iba slabé alebo žiadne prepojenia. Riešenie prináša sémantizácia webu. Sémantizácia existujúcich údajov, uložených v digitálnych knižniciach, môže priniesť nové poznatky a informácie.

1.2 Informácie v digitálnych knižniciach

Knižnice patrili od samého začiatku k inštitúciám, ktoré zhromažďovali, spracovávali a poskytovali informácie. S týmto cieľom si vytvárali vlastné katalogizačné a výmenné pravidlá, ktoré umožňovali efektívnu výmenu informácii. Rozvojom technológií ovplyvnil klasické knižnice a spôsob akým spracovávajú a poskytujú informácie svojim používateľom. Rozvoj internetu podnietil vznik digitálnych knižníc a knižničných systémov.

Digitálne knižnice môžeme definovať ako online zbierku digitálnych objektov zaručenej kvality, vytvorenú a spravovanú podľa medzinárodne platných zásad budovania zbierok.[1]

Pod pojmom digitálny objekt sa rozumie elektronický časopis, elektronická kniha, konferenčné materiály a multimediálne dokumenty (zvukové, obrazové, audiovizuálne dokumenty a počítačové programy).

Digitálne knižnice taktiež obsahujú bibliografické údaje, ktoré poskytujú základné informácie o existencii dokumentu, jeho obsahu príp. určujú jeho miesto v systéme dokumentov a poznatkov. Bibliografické údaje sú v bibliografickom popise usporiadané logicky v rámci bibliografického záznamu, ktorý môže obsahovať aj ďalšie údaje, a ktorý zastupuje dokument alebo dokumenty v rozličných komunikačných situáciách.

Bibliografický záznam obsahuje vo svojej opisnej časti údaje o autorovi, názvov diela, dátum a miesto vydania, informácie o vydavateľovi, rozsahu, obsahu, jazyku dokumentu a ďalšie identifikačné údaje. Bibliografický záznam môže obsahovať aj rôzne klasifikačné prvky, ktoré umožňujú tematické zaradenie dokumentu.

Všetky bibliografické údaje v digitálnych knižniciach sú uchovávané a zdieľané pomocou vlastného dátový formátu (MARC) a zoznamu pravidiel určujúci obsah a formu záznamov (AACR, ISBD).

[1] IFLA. *IFLA/UNESCO Manifesto for Digital Libraries,* s.1

1.2.1 MARC

Prvým široko používaným formátom na výmenu bibliografických údajov sa stal MARC, ktorý vznikol v šesťdesiatych rokoch minulého storočia v Kongresovej knižnici (USA).
Vylepšenou verziou MARC-u sa stal MARC 21, ktorý vznikol spojením amerického a kanadského formátu (USMARC a Canadian MARC). MARC 21 (*MARC for 21st century*) umožňuje komunikáciu a výmenu bibliografických dát bez ohľadu na programové vybavenie používateľov. Formát MARC 21 umožňuje spracovanie a výmenu bibliografických dát medzi knižnicami a informačnými subjektmi na národnej alebo medzinárodnej úrovni.

Aktuálna verzia MARC 21 sa skladá z prvku *leader*, ktorý obsahuje všeobecné informácie (napr.: dĺžku záznamu) a polí obsahujúcich bibliografické údaje. Každé pole je označené značkou (tagom) v rozsahu od 001 po 999. Nie všetky polia sú využité. Polia s tagom 001-008 sa nazývajú variabilné riadiace polia (*controlfields*), zvyšné polia sa nazývajú variabilné polia údajov (*datafields*).

Polia s hodnotou väčšou ako 008 sa vo formáte MARC 21 delia na podpolia. Podpolia sú oddelené oddeľovačom. V MARC 21 sa najčastejšie používa znak dolára $, ale môžu byť použité aj iné znaky. Každému podpoľu sa priraďuje písmeno alebo číslica. Okrem toho sa môžu k poliam väčším ako 010 pridávať dva pomocné indikátory. Obidva indikátory sú navzájom nezávislé a označujú sa číslami 0 až 9.

Variabilné polia sú zoskupené do blokov podľa prvého čísla v tagu, ktorý, až na niektoré výnimky, identifikuje funkcie údajov v zázname.

Typ informácie v poli je identifikovaný číslami v zostatku tagu:

0XX Riadiace informácie, identifikačné a klasifikačné čísla atď.

1XX Hlavné vstupy

2XX Blok popisných informácií (názov, vydavateľské údaje, údaje o vydaní, poradí vydania)

3XX Fyzický popis

4XX Údaje o edícii

5XX Poznámky

6XX Polia prístupu podľa predmetu

7XX Pridané vstupy iné ako podľa predmetu alebo edície, polia väzieb

8XX Pridané vstupy o edíciách, holdingoch atď.

9XX Rezerva pre lokálnu implementáciu

Okrem formátu MARC 21 je v súčasnosti rozšírený aj formát UNIMARC (UNIverzal MARC formát), ktorý bol predstavený organizáciou IFLA v roku 1977. Aktuálna tretia verzia UNIMARCu bola publikovaná v 2008.

Ukážka štruktúry MARC 21 záznamu, ktorý bol získaný z knižničného informačného systému Virtua Slovenskej národnej knižnice[2]:

Pole	Ind. 1	Ind. 2	Dáta
001			vtls010170925
003			VRT
005			20120225103100.0
008			100922s2010 xo g 000 f slo
015			\a SNBA2011/01-00115
020			\a 9788055122502 (viaz.)
039		9	\a 201202251031 \b mbpksh04 \c 201112061247 \d mpppk02 \c 201103031206 \d msnk36 \c 201011251433 \d msavuk02 \y 201009221332 \z msnk99
040			\a SNK \b slo \c SNK \e AACR2
041	1		\a slo \h por
044			\a xo \c SK
072		7	\a 821.134.3 \x Portugalská literatúra \2 konspekt
080			\a 821.134.3(81)-31 \2 2001
080			\a 929 \2 2001
080			\a 27-244 \2 2001
080			\a 27-236.5 \2 2001
080			\a (0:82-312.6) \2 2001
100	1		\a Coelho, Paulo, \d 1947-
240	1	0	\a O monte cinco \l Slovensky
245	1	0	\a Piata hora \c Paulo Coelho ; [preklad: Miroslava Petrovská]
250			\a 3. vyd.
260			\a Bratislava \b Ikar \c 2010
300			\a 214 s. \c 21 cm
600	0	7	\a Eliáš, \c prorok, \d 10 stor. pr. Kr. \2 SNKPH
630	0	7	\a Biblia \p S.Z. \p Kráľovská, 1. \2 SNKPH
650	0	7	\a starozákonní proroci \2 SNKPH
650	0	7	\a biblické príbehy \2 SNKPH
655		7	\a brazílske romány \2 SNKPH
655		7	\a biografické romány \2 SNKPH
700	1		\a Petrovská, Miroslav \4 trl
919			\a 978-80-551-2250-2

[2] https://www.kis3g.sk/

1.2.2 Vyhľadávanie v digitálnych knižniciach

Rozvoj internetu a webu spôsobuje zmeny v spôsobe vyhľadávania informácií. Knižnice strácajú svoje dominantné postavenie v oblasti poskytovania informácií. Konkurenciou sa stali webové vyhľadávacie služby ako Google, Amazon, Yahoo, atď (1).

Viaceré vyhľadávacie služby poskytujú okrem hľadaných informácii aj ďalšie súvisiace informácie a odkazy k ďalším zdrojom, čo uľahčuje vyhľadávanie. V porovnaní so spomenutými službami neposkytujú online knižničné katalógy (OPAC) dostatočný používateľský komfort. Vyhľadávanie v katalógoch je často krát náročné, pretože používateľ musí vyhľadávať pomocou formulárov a vyberať správne kategórie vyhľadávania, aby sa dopracoval k želaným výsledkom. Ďalšou nevýhodou vyhľadávania v knižniciach môže byť obmedzenie na fond jednej knižnice. Bibliografické záznamy nemajú priame prepojenia na ďalšie alebo súvisiace záznamy. Chýbajú tu sémantické väzby medzi dokumentmi a inými informačnými zdrojmi.

Objavujú sa preto rôzne iniciatívy, ktorých cieľom je zefektívniť vyhľadávanie informácií v knižniciach a zvýšiť interoperabilitu medzi knižničnými systémami a online dostupnými informačnými zdrojmi.

Jednou z iniciatív je zapojenie knižníc do projektu linked data a sprístupnenie bibliografických údajov na webe. Publikovanie bibliografických dát vo forme linked data umožňuje vytvoriť nový spôsob navigácie informačným knižničným fondom a efektívnejšie vyhľadávanie v ňom. Nové rozhranie môže poskytnúť prístup obohatený o kontextové informácie. Transformácia bibliografických dát s použitím URI odkazov umožní zhmotniť väzby, ktoré sa nachádzajú v bibliografických dátach, no nie sú dobre vyznačené (napr. medzi bibliografickým záznamom a odkazom na záhlavie autority).

Ďalšou iniciatívou je vznik Funkčných požiadaviek y na bibliografické záznamy (*Functional Requirements for Bibliographic Records - FRBR*). FRBR predstavuje nový konceptuálny entitno-vzťahový model vytvorený organizáciou IFLA (*International Federation of Library Associations and Institutions*), ktorý slúži používateľom na vyhľadávania a prístup do on-line katalógov knižníc a bibliografických databáz. Významnou črtou modelu je jeho nezávislosť od konkrétnych katalogizačných štandardov ako je AACR2 alebo ISBD. FRBR definuje základné entity, ich vlastnosti ako aj vzájomné vzťahy týchto entít. Jeho cieľom je vyhľadať, identifikovať, vybrať a získať entity.

Rovnako vznikol aj nový štandard *Resource Description and Access* (RDA), ktorý nahrádza katalogizačné pravidlá AACR. Do vývoja boli zapojené viaceré komunity (archívy, vydavatelia, múzeá, atď.) v snahe dosiahnuť efektívne prepojenie medzi RDA a metadátovými štandardami, ktoré tieto komunity používajú.

Objavili sa diskusie a názory, ktoré tvrdia, že MARC formát je vhodný na reprezentáciu a výmenu informácií v rámci digitálnych knižníc, no nie je dostatočne flexibilný pre potreby súčasného webu.[3] Viaceré inštitúcie a knižnice sa zapojili do projektov, ktoré prevádzajú bibliografické údaje do alternatívnych formátov vyhovujúcich súčasným potrebám.

1.3 Projekty sémantizácie bibliografických údajov

V tejto kapitole sú spomenuté niektoré významné projekty a iniciatívy zamerané na sémantizáciu bibliografických dát a ich prepojenie s webovým prostredím. Do projektov sa zapojili prevažne národné a univerzitné knižnice rôznych krajín.

[3] TENNANT, R. *MARC Must die*, s. 28

1.3.1 BIBFRAME

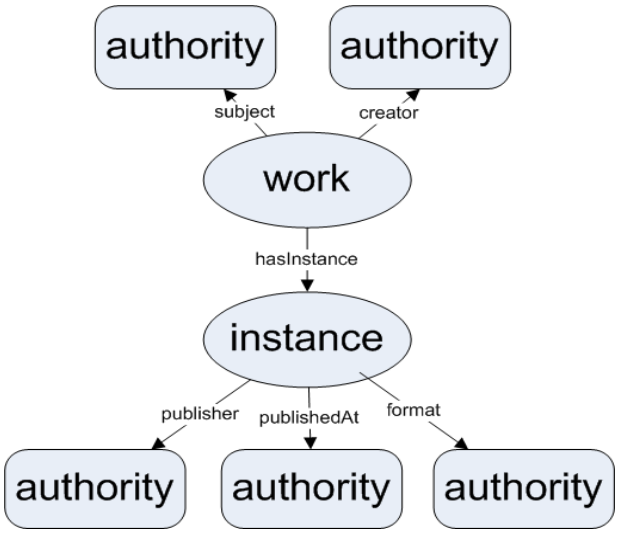
V máji 2011 spustila Kongresová knižnica projekt s názvom Bibliographic Framework Initiative, ktorého cieľom je vytvoriť nový model bibliografických dát. Model dostal meno BIBFRAME (skratka zo slov BIBliographic FRAMEwork).

Kongresová knižnica stojí úspechom modelu/formátu MARC, ktorý sa stal od šesťdesiatych rokov najpoužívanejším formátov v digitálnych knižniciach a príbuzných organizáciách. Len Kongresová knižnica a Britská knižnica majú okolo 10 miliónov záznamov v MARC formáte[4]. Formát bol počas svojej viac ako 40-ročnej histórie postupne viackrát upravovaný a optimalizovaný.

Aj napriek všetkému úsiliu už MARC formát nevyhovuje súčasným potrebám. Kongresová knižnica vidí potrebu vytvoriť model bibliografických dát, ktorý bude poskytovať novú reprezentáciu postavenú na jednoznačne identifikovaných entitách a vzťahoch medzi nimi. Cieľom knižnice je integrovať svoje dáta s Linked Open dátami a vytvoriť tak širokú bázu dát, ktorá pomôže používateľom pri vyhľadávaní relevantných informácií.

BIBFRAME model sa skladá z nasledujúcich tried:

- **Práca (Work)** - je zdroj odrážajúci konceptuálnu podstatu katalogizačnej položky (dielo, ideový obsah)
- **Inštancia (Instance)** - je fyzické, materiálne vyjadrenie diela
- **Autorita (Authority)** - zdroj, ktorý zahŕňajú osoby, miesta, organizácie, témy, atď.
- **Anotácia (Anotation)** - obohacuje predchádzajúce zdroje o ďalšie prídavné informácie (napr.: holdingové informácie, hodnotenia, webové stránky, informácie o prevedení diela - obálka, rozmery)



Obrázok č. 1 Model BIBFRAME[5]

Model BIBFRAME definuje entity, atribúty a vzťahy medzi entitami. Využíva RDF formát zápisu, pomocou ktorého môže reprezentovať všetky entity (zdroje), atribúty a vzťahy medzi entitami ako jednoznačné webové zdroje.

Na domovskej stránke projektu[6] sa nachádzajú podrobnejšie informácie o modeli, všetkých vzťahoch a vlastnostiach, ktoré môžu nadobúdať entity, URI identifikátory vlastností, jednotlivé polia a podpolia MARC záznamu využívané pri napĺňaní modelu.

Súčasťou sémantizácie MARC záznamov v Kongresovej knižnici je vyhľadanie a priradenie URI identifikátora osoby (autora, vydavateľa, organizácie), ktoré čerpá z Medzinárodný virtuálny súbor autorít VIAF.

---

[4] STYLES, R., AYERS, D., SHABIR, N. *SEMANTIC MARC, MARC21 AND THE SEMANTIC WEB,* s. 1
[5] MILLER, E., OGBUJI, U., MUELLER, V., MACDOUGALL, K. *Bibliographic Framework as a Web of Data: Linked Data Model and Supporting Services,* s. 9
[6] BIBFRAME.ORG. *Model Overview.* [Online]

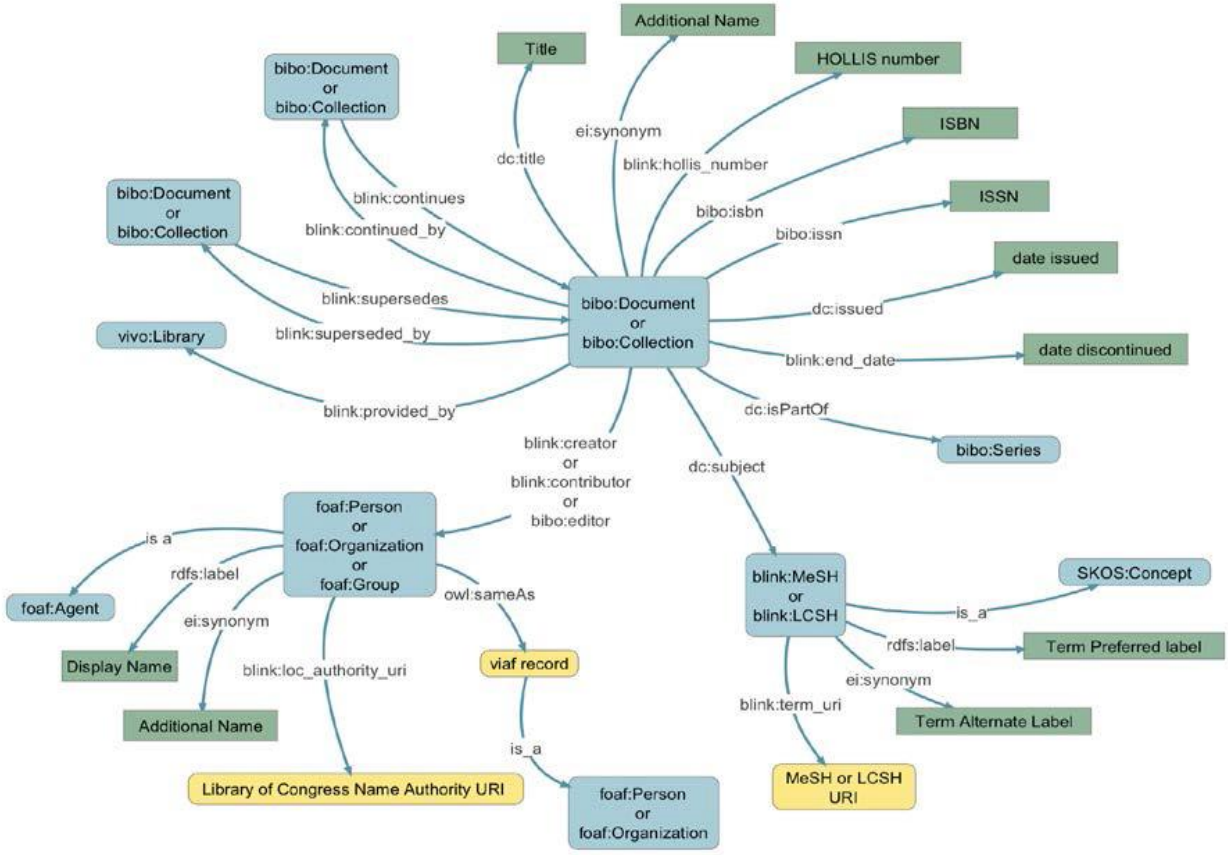
1.3.2 BLINK

Projekt Blink vznikol na pôde Countway Digital Library. Digitálna knižnica ktorá sa skladá z elektronických kníh, časopisov a databáz, ktoré súvisia so záujmami Harvardskej lekárskej fakulty, jej výskumných pracovníkov a študentov.

Každý mesiac využíva prístup k elektronickým materiálov knižnice viac ako 30.000 čitateľov. Konfigurácia a nastavenie digitálnej knižnice poskytovali obmedzené funkcie vyhľadávania, navyše s rastúcim objemom digitálnych materiálov knižnica stále častejšie narážala na problém s efektívnym a korektným vyhľadávaním. Vedenie knižnice sa rozhodlo zmeniť nevyhovujúcu situáciu a vytvoriť projekt Blink.

Úlohou projektu je stransformovať existujúce bibliografické údaje do podoby Linked Open Data. Nová reprezentácia bibliografických údajov pomáha zvyšovať presnosť vyhľadávania a umožňuje prepojenie s ďalšími zdrojmi otvorených dát, ako napríklad Harvard Catalyst's Profiles alebo VIAF.

Projekt využíva vlastnú ontológiu s názvom *blink*. Ontológia využíva mnohé entity a vzťahy prebraté z ontológii BIBO, Dublin Core, SKOS, VIVO a foaf.[7]
Model blink ontológie je znázornená na obrázku č.2.



Obrázok č. 2 Model Blink ontológie[8]

[7] CHENG, S. *Linked Open Data for Countway Library Final report for Phase 1,* s.1
[8] Cit. 7, s. 2

V procese transformácie sa venuje zvýšená pozornosť poliam, v ktorých je uložená informácia o predmete (*subject*). Text z týchto polí sa nahrádza URI identifikátorom, ktorý sa hľadá v slovníkoch a predmetových heslách:

- NCBI's MeSH - Medical Subject Headings - kontrolovaný tezaur databázy PubMed
- LCSH - predmetové heslá Kongresovej knižnice

Vzniknuté záznamy sa následne mapujú do blink ontológie. Úplný zoznam polí a podpolí, ktoré sa extrahujú pri transformácii MARC záznamu spolu so spôsobom ich namapovania do blink ontológie je možné nájsť v dokumentácii projektu Blink.[9]

1.3.3 BNB

Britská národná knižnica vytvára súbor britskej národnej bibliografie vo forme Linked Open Data. Súbor zahŕňa všetky publikované knihy a časopisy. Súbor bol vytvorený približne z 2.8 milióna záznamov, ktoré vyprodukovali 89 733 617 RDF trojíc.[10]

Iniciatíva Britskej národnej knižnice sa líši od ostatných knižníc, ktoré iba transformujú MARC záznamy a vytvoria novú reprezentáciu dát. Pri transformácií MARC záznamov BNB upriamila svoju pozornosť na tzv. veci záujmu (*things of interest*), ako sú ľudia, miesta, témy a udalosti spojené s knihou alebo publikáciou.
Veci záujmu sú modelované pomocou existujúcich ontológií a slovníkov:
BIBO, Bio, Dublin Core, Event Ontology, FOAF, ISBD, Org: An Organization Ontology, OWL, SKOS, RDF Schema, Geo Positioning a RDA. Existujúce ontológie nepokrývajú všetky potrebné triedy a vlastnosti na modelovanie vecí záujmu, preto si BNB vytvorila vlastnú ontológiu s názvom British Library Terms (BLT).[11]

BNB prepája svoje záznamy s externými zdrojmi:

- VIAF - Medzinárodný virtuálny súbor autorít osobných mien
- LCSH - Predmetové heslá Kongresovej knižnice
- Lexvo - informácie o jazyku a písme
- GeoNames - krajiny a mestá
- MARC - identifikačné kódy krajín
- Dewey.info - mapovanie Deweyho desatinného triedenia

Vytvorený model[12] je stále v procese vývoja a môže sa meniť.

1.3.4 DNB Linked Data Service

Od roku 2010 poskytuje Nemecká národná knižnica (*Deutsche Nationalbibliothek*) svoje bibliografické údaje vo forme Linked Data. Jej cieľom je zvýšiť použiteľnosť a dostupnosť bibliografických údajov a pritiahnuť pozornosť nových skupín používateľov, ako sú vydavatelia, prevádzkovatelia vyhľadávačov

[9] CHENG, S. *Linked Open Data for Countway Library Final report for Phase 1*, s. 8-11
[10] THE BRITISH LIBRARY. Metadata Service. [Online]
[11] http://www.bl.uk/schemas/bibliographic/blterms
[12] THE BRITISH LIBRARY. *British Library Data Model - Book.* [Online]

a systémov riadenia vo výskumných inštitúciách a neziskových organizáciách. DNB skúma potreby týchto skupín a snaží sa prispôsobovať svoje služby.

Okrem bibliografických dát sprístupnila DNB aj súbory autorít vo forme Linked Data:
PND (osobné mená autorov), SWD (predmetové heslá) a GKD (názvy korporácií).

Všetky údaje poskytované touto službou sú exportované vo formáte RDF/XML a RDF/turtle. Exportované záznamy RDF nemajú rovnakú zložitosť ako pôvodné MARC záznamy. Pri exporte sa používajú iba niektoré významné polia.

Nový dátový model je vytvorený pomocou viacerých široko používaných ontológií sa slovníkov:
RDA Element Set, Relationship Vocabulary, SKOS, Bibo, Dublin Core, RDF Schema, OWL a FOAF. DNB si vytvorila vlastný slovník - GND (*Gemeinsame Normdatei*), na modelovanie špeciálnych vlastností, ktoré nie sú obsiahnuté v žiadnej z použitých ontológií.
DNB využíva viaceré externé zdroje:

- DDC - Nemeckú verziu Deweyho desatinného triedenia
- LCSH - Predmetové heslá Kongresovej knižnice
- Rameau - Predmetové heslá Francúzskej národnej knižnice
- DBpedia
- VIAF

1.3.5 Ďalšie projekty

K projektom sémantizácie zaraďujeme aj Libris - projekt Švédskej národnej knižnice, *Biblio* - projekt Univerzitnej knižnice v Gente[13], projekt EPrints - Fakulty elektroniky a informatiky Univerzity v Southampton vo Veľkej Británii[14], FRBR[15], Biblioteca Nacional de España - MARiMbA[16] a iné. Ich procesy a výstupy sú podobné ako v predchádzajúcich projektoch, preto nie sú podrobnejšie opisované v tejto práci.

1.4 Ontológie v informatike

Ontológia v informatike je spôsob reprezentácie znalostí o svete alebo jeho časti. Je to dátový model, ktorý reprezentuje množinu pojmov a vzťahy medzi nimi. Z explicitne vyjadrených znalostí zaznamenaných v ontológii možno vyvodzovať implicitné dôsledky a súvislosti zahrnuté v ich obsahu.

Ontológia definuje slovník pojmov, ich vlastnosti a vzťahy medzi nimi. Poskytuje slovník termínov a gramatiku, ktorá obsahuje pravidla pre ich vytváranie. Ontológia zavádza množstvo štruktúrnych a konceptuálnych vzťahov vrátane vzťahu nadtrieda/podtrieda/inštancia, vzťahy k času. Podporuje taktiež terminologické odvodzovanie. Obsahuje prostriedky na rozlišovanie sémanticky odlišných a naopak združovanie sémanticky blízkych termínov. Vďaka formálnemu zápisu rozumejú ontológiám aj stroje, čo otvára ďalšie možnosti spracovania dát.

V oblasti digitálnych knižníc a bibliografických databáz sa používa niekoľko ontológií, ktoré sa dajú použiť samostatne alebo v rôznych kombináciách. Medzi najrozšírenejšie ontológie z tejto oblasti patria BIBO, FaBiO, SKOS a CiTO.

[13] https://biblio.ugent.be/input
[14] http://eprints.ecs.soton.ac.uk/
[15] http://www.ifla.org/publications/functional-requirements-for-bibliographic-records
[16] BIBLIOTECA NACIONAL DE ESPANA. *MARiMbA: conversión de MARC 21 a RDF*. [Online]

BIBO - bibliografickú ontológiu vytvorili D'Arcus a Frédérick Giasson. Je to prvá OWL Full ontológia, ktorá slúži na opis bibliografických údajov v prostredí sémantického webu. Hlavnou triedou celého modelu je *bibo:Document*. BIBO obsahuje iba 69 tried a 106 vlastností. Ontológia môže byť použitá ako citačná ontológia, ontológia na klasifikáciu dokumentov alebo jednoducho ako spôsob zápisu ľubovoľného dokumentu vo formáte RDF. BIBO ontológia bola inšpirovaná mnohými existujúcimi formátmi opisujúcimi bibliografické metadáta. BIBO môže byť spájaná a rozširovaná ďalšími slovníkmi a ontológiami, napríklad Dublin Core, PRISM a FOAF na opis autorít. Ontológia je navrhnutá tak, aby mohla byť v budúcnosti ľahko rozširovateľná o ďalšie moduly[17]. V súčasnosti obsahuje niekoľko tried a vlastností, ktoré umožňujú reprezentovať publikačnú doménu (*bibo:AcademicArticle, bibo:Journal, bibo:Collection, bibo:Book,bibo:Chapter a bibo:Issue*).[18]

FaBiO - je ontológia na reprezentáciu a publikovanie popisov entít na sémantickom webe, ktoré sú publikované alebo potenciálne publikovateľné, a ktoré obsahujú alebo sa odvolávajú na bibliografické citácie, alebo subjekty používané na definovanie týchto bibliografických citácií. FaBiO entity sú predovšetkým textové publikácie ako napríklad knihy, časopisy, noviny a predmety ich obsahu. Taktiež môžu obsahovať súbory dát, počítačové algoritmy, experimentálne protokoly, formálne špecifikácie a slovníky, právne záznamy, vládne dokumenty, technické a obchodné správy, referenčné zoznamy, katalógy knižníc a podobné zbierky. S použitím ontológie FaBiO môžeme zachytiť výskumnú činnosť ako *fabio:ResearchPaper* (dielo - work), ktorý môže byť publikovaný ako *fabio:JournalArticle*, prípadne *fabio:ConferencePaper* alebo *fabio:BookChapter* (možné vyjadrenia diela). BIBO obsahuje triedu *bibo:AcademicArticle,* ktorá zahŕňa obidva koncepty *fabio:ResearchPaper* a *fabio:JournalArticle.* FaBiO triedy sú usporiadané podľa FRBR schémy dielo, výraz, prejav a položka[19]. FaBiO obsahuje 211 tried a 123 vlastností.

Obidve ontológie poskytujú možnosti na opis a reprezentáciu bibliografických údajov, pritom sa však líšia v štruktúre a iných vlastnostiach, ktoré treba zvážiť pri výbere jednej z nich.

CiTO - Citation Typing Ontology je ontológia na opis referenčných citácií vo vedeckých výskumných článkoch, ďalších odborných prácach a tiež webových informačných zdrojoch. CiTO môže byť použité na publikovanie citácií na sémantickom webe. CiTO je tiež súčasťou SPAR ontológií a formálne preberá konceptuálny model FRBR. Obsahuje podtriedy *cito:Work*, *cito:Expression* a *cito:Manifestation,* ktoré umožňujú presnejšie vyjadrenie citovaných zdrojov. CiTO obsahuje 23 vzťahov na vyjadrenie citácií, ktoré sa rozdeľujú do dvoch hlavných skupín: Factual relationships a Rhetorical relationships. Factual relationships obsahujú vzťahy na citovanie diel, autorít, použitých metód alebo dát (*cito:cites, cito:isCitedBy, cito:usesMethodIn, cito:citesAsSourceDocument* a iné). Rhetorical relationships obsahuje vzťahy, ktoré umožňujú vyjadrovať súhlas alebo nesúhlas s citovaným dielom (*cito:supports, cito:confirms, cito:disagreesWith, cito:critiques* a iné). BIBO tiež obsahuje niekoľko vzťahov na vyjadrenie citácií (*bibo:cites* , *bibo:affirmedBy*, *bibo:annotates*, *bibo:reviewOf* a *bibo:translationOf*), ale v porovnaní s CiTO ontológiou má slabšiu citačnú schopnosť.[20]

[17] The Bibliographic Ontology. *Bibliographic Ontology Specification.* [Online]
[18] PERONI, S., SHOTTON, D. *FaBiO and CiTO: Ontologies for describing bibliographic resources and citations,* s. 35
[19] Cit. 19, s. 38
[20] SHOTTON, D., *CiTO, the Citation Typing Ontology,* s. 40

FOAF - Ontológia poskytujúca triedy a relácie na opis ľudí, objektov, aktivít a vzťahov medzi nimi. Dovoľuje skupine ľudí vymodelovať sociálnu sieť a sledovať spojenia a vzťahy medzi osobami.

Dublin Core - je súbor metadátových prvkov, ktorých úlohou je uľahčiť vyhľadávanie elektronických zdrojov. Vďaka svojim vlastnostiam ako jednoduchosť, modularita, rozšíriteľnosť zaujal inštitúcie zaoberajúce sa formálnym spracovávaním zdrojov, ako sú knižnice, múzeá, vládne agentúry či komerčné organizácie. Súbor Dublin Core sa skladá z 15 prvkov, ktoré umožňujú jednoduché a komplexné vyjadrenie obsahu zdroja. V časti príloha A sa nachádza kompletný zoznam prvkov spolu s ich popisom.

SKOS - Za odľahčený ontologický jazyk sa dá považovať SKOS, ktorý je prispôsobený pre reprezentáciu znalostných systémov ako sú tezaury, riadené slovníky, hesláre alebo systematické klasifikácie.
Veľké množstvo známych tezaurov a klasifikačných systémov začalo prevádzať svoje údaje do SKOS dokumentov. Tým sa SKOS stal štandardom pre kódovanie riadených slovníkov pre vytváranie sémantického webu.

Pre oblasť vedy a výskumu existuje niekoľko vlastných ontológií, ktoré sa snažia zachytiť vzťahy medzi výskumníkmi, výskumnými oblasťami, organizáciami a školami. Medzi používané a progresívne ontológie zaraďujeme VIVO, SWRC a CERIF štandard.

VIVO je ontológia reprezentujúca akademické výskumné komunity. VIVO je súčasne aj webový informačný systém, na napĺňanie a prezentovanie údajov, uložených v tejto ontológií. Každá osoba, organizácia alebo iný subjekt vo VIVO má priradený jedinečný identifikátor vo forme URI. Vďaka tomu VIVO umožňuje objavovať výskumníkov naprieč rôznymi inštitúciami. Ontológia poskytuje sadu tried a vzťahy (vlastnosti) sa predstavujúce výskumníkov a širší kontext, v ktorom pracujú. Obsah lokálnych VIVO ontológií môže byť napĺňaný ručne používateľmi alebo automatizovaným spôsobom z rôznych zdrojov a databáz.

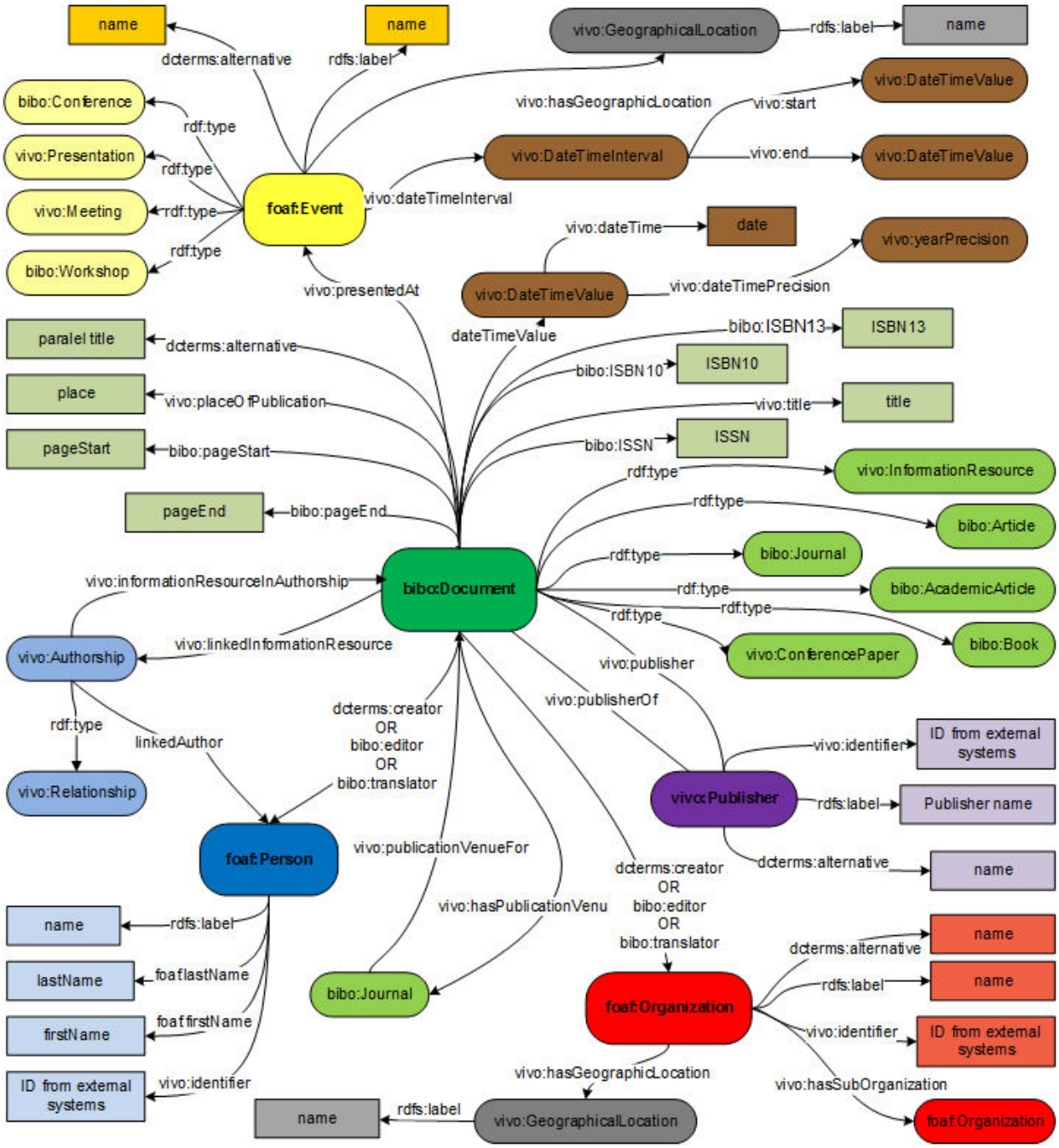
VIVO používa 7 inštitúcií, ktoré sa zapojili do grantu NIH (*National Institutes of Health*) s cieľom vytvoriť národnú sieť výskumníkov. VIVO sa používa napríklad na Cornell University (zakladateľská inštitúcia), The Scripps Research Institute, University of Florida alebo University of Melbourne. Do projektu sa súčasne zapája 50 univerzít a výskumných inštitúcií po celom svete.[21]

Pri vývoji VIVO ontológie sa kládol dôraz na zabezpečenie interoperability s ďalšími heterogénnymi ontológiami. Ontologická interoperabilita sa dosahuje identifikáciou a vytvorením vzťahov medzi zdrojmi a vlastnosťami jednotlivých ontológií. Súčasná verzia VIVO obsahuje nasledovné ontológie a slovníky: Dublin core, Event ontology, FOAF, Geopolitical ontology, SKOS, BIBO a OWL.

1.4.1 VIVO model

Na obrázku č. 3 sa nachádza časť VIVO modelu (triedy, atribúty a väzby), ktoré je možné naplniť dátami z bib. záznamov. Celý VIVO model je omnoho rozsiahlejší.[22]

[21] VIVO. *VIVO Ontology.* [Online]
[22] VIVO. *VIVO model.* [Online]



Obrázok č. 3 Vybrané triedy a vzťahy vo VIVO modeli

Vybranú časť modelu tvorí 5 entít:

Document dielo vytvorené, preložené alebo upravené osobou alebo korporáciou. Dielo môže nadobúdať niektorú z foriem: článok(*Article*), kniha(*Book*), žurnál (*Journal*), atď.

Person osoba (autor, prekladateľ, editor).

Organization korporácia

Publisher vydavateľ

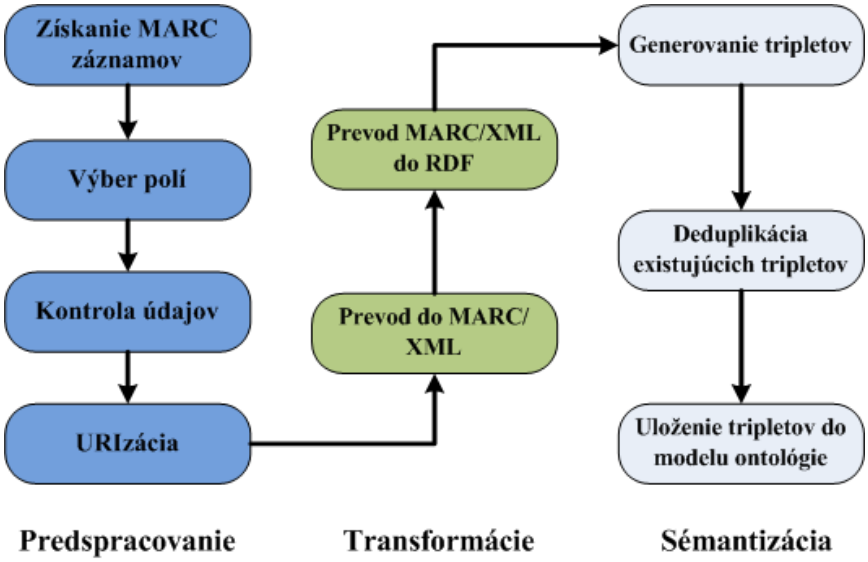
Event akcia. Akcia. Napr.: konferencie (*Conference*), stretnutia, (M*eeting*), atď.

1.5 Proces sémantizácie bibliografických dát

V ďalšej časti dokumentu sú analyzované procesy a postupy, ktoré sa používajú pri transformácií bibliografických dát na ontológie.

1.5.1 Proces transformácie MARC záznamov

Proces sémantizácie bibliografických záznamov je približne rovnaký vo všetkých spomenutých knižniciach. Rozdiely súvisia s kvalitou bibliografických dát, ich úplnosťou, externými zdrojmi, ktoré sú použité pri identifikácií autorít a ontológií do ktorých sú dáta mapované. Na obrázku č. 4 je znázornený proces sémantizácie MARC záznamov.



Obrázok č. 4 Proces transformácie a sémantizácie MARC záznamu.

Podrobnejší opis procesu:

Predspracovanie - proces predspracovania sa skladá z viacerých krokov:

1. **získanie MARC záznamov**
2. **výber polí a podpolí** - Obsah polí a podpolí závisí od použitých katalogizačných pravidiel a MARC formátu. V nasledujúcej kapitole sú bližšie uvedené niektoré polia MARC 21 spolu s bibliografickými údajmi, ktoré sú v nich uložené.
3. **kontrola úplnosti vybraných údajov** - záznamy, ktoré nespĺňajú minimálne požiadavky na spracovanie sú vyradené z procesu, prípadne sú poslané na korekciu.
4. **URIzácia** - vytvorenie alebo vyhľadanie existujúceho lokálneho URI identifikátora pre vybranú autoritu. V tomto procese sa vyhľadávajú aj URI identifikátory v externých zdrojoch a databázach (napr.: VIAF, LCSH, SKOS, atď.) V prípade, že knižnica nepoužíva národný súbor autorít, musí pri prideľovaní URI identifikátora rozlišovať/identifikovať autority (autorov, vydavateľov, organizácie, názvy miest). Najčastejším problémom je identifikácia autorov s rovnakým menom alebo autorov, ktorí majú preklep v mene. Problémom sú aj variantné formy názvu organizácií (napr.: UMB, Univerzita Mateja Bela).

Transformácia - Transformované záznamy nadobúdajú novú reprezentáciu. V tejto fáze ešte nemajú pridanú sémantickú hodnotu.

5. **prevod z MARC formátu do MARC/XML** - mnohé knižničné systémy dokážu exportovať záznamy v tejto forme.
6. **prevod MARC/XML do RDF** - využívajú sa viaceré nástroje ako napríklad MARiMbA[23] alebo MARC/MODS RDFizer[24].

Sémantizácia - vytváranie sémantických vzťahov

7. **Generovanie tripletov** - dochádza k vytváraniu RDF trojíc a mapovaniu na vybrané ontológie.
8. **Deduplikácia existujúcich tripletov**
9. **Uloženie tripletov do modelu ontológie** - finálna fáza transformácie. Triplety sa stávajú dostupné pri vyhľadávaní.

Knižničné dáta obsahujú veľké množstvo metadát, ktoré nemajú pre používateľa ani proces sémantizácie význam. V nasledujúcej časti dokumentu sa nachádza zoznam MARC polí, medzi ktorými sa dajú identifikovať sémantické väzby.

1.5.2 Výber MARC polí

Údaje v knižničných katalógoch majú presne stanovenú štruktúru a pravidlá zapisovania bibliografických dát (AACR, ISBD). Ako bolo spomenuté v predchádzajúcich častiach dokumentu, knižničné systémy využívajú rôzne formáty, ktoré slúžia na zobrazenie a výmenu bibliografických údajov. V nasledujúcej časti dokumentu sa nachádza zoznam bibliografických údajov zapísaných vo formáte MARC 21, ktoré ponúkajú možnosti na naplnenie VIVO ontológie.

Osobné meno

Osobné meno (hlavný vstup) sa nachádza v poli 100. Zvyčajne označuje osobu alebo osoby, ktoré majú hlavnú zodpovednosť za dielo. Aj v poli 700 sa nachádza osobné meno (pridaný vstup), ktorý určuje vedľajšiu zodpovednosť (napr. prekladateľ, ilustrátor).

Meno a priezvisko sa v MARC 21 zapisuje do jedného spoločného podpoľa. Ako oddeľovač sa používa čiarka. V prípade UNIMARCu sa meno a priezvisko zapisuje do samostatných podpolí.

Pole	Podpole	poznámka
100	a	Meno autora
	c	Tituly a iné slová súvisiace s menom
	d	Podpole $d obsahuje dátumy narodenia, úmrtia alebo tvorivého obdobia, alebo iný dátum použitý spolu s menom.
700	a	Meno autora
	c	Tituly a iné slová súvisiace s menom
	d	Dátumy súvisiace s menom
	4	Kód roly. Napr.: aut - autor, trl - prekladateľ

[23] ONTOLOGY ENGINEERING GROUP. *MARiMbA*. [Online]
[24] SIMILE. *MARC/MODS RDFizer*. [Online]

Ukážka:

100 1 # \a Freud, Sigmund \d 1856 - 1939.

700 1 # \a Bieliková, Mária, \d 1966- \4 edt

Meno korporácie

Pole 110 a 710 obsahujú korporatívne meno. Aj tu platia rovnaké pravidlá o primárnej a vedľajšej zodpovednosti ako pri osobných menách.

Pole	Podpole	poznámka
110	a	Meno korporácie alebo organizácie
710	a	Meno korporácie alebo organizácie
	4	Kód roly. Napr.: aut - autor, trl - prekladateľ

Ukážka:

710 2 # \a Indian Council for Research on International Economic Relations

Meno združenia

Pole 111 a 711 obsahujú meno združenia. Aj tu platia rovnaké pravidlá o primárnej a vedľajšej zodpovednosti ako pri osobných menách.

Pole	Podpole	poznámka
111	a	Meno korporácie alebo organizácie
	c	Miesto konania zhromaždenia
	d	Dátum konania zhromaženia
711	a	Meno korporácie alebo organizácie
	c	Miesto konania zhromaždenia
	d	Dátum konania zhromaženia
	4	Kód roly. Napr.: aut - autor, trl - prekladateľ

Ukážka:

111 2 # \a Student Research Conference \n (8. : \d 2012 : \c Bratislava, Slovensko)

Vydavateľské údaje

Pole 260 obsahuje informácie súvisiace s publikovaním, tlačou, produkciou a rozširovaním, diela. Ak ide o nepublikované dokumenty, toto pole nemusí byť zahrnuté v zázname, prípadne môže obsahovať len podpole $c (Dátum publikovania, distribúcie atď.).

Pole	Podpole	poznámka
260	a	Miesto publikovania, distribúcie atď.
	b	Názov vydavateľa, distibútora atď.
	c	Dátum publikovania, distribúcie atď.

Ukážka:

260 \a Bratislava \b Slovak University of Technology \c 2012

Dátum publikovania, distribúcie (260 \c) sa nachádza aj v poli 008 na pozícií 7-10.

*008 120509s**2012** xo dal f 101 0 eng*

Názov diela

Názov diela sa nachádza v poli 245. Toto pole obsahuje okrem hlavného názvu aj zvyšok názvu, ďalšie názvové informácie a údaje o zodpovednosti. Hlavný názov obsahuje skrátený názov a alternatívny názov, číselné označenie časti a názov časti.

Ak má dielo meniace sa formy názvu, ktoré sa podstatne odlišujú od názvu v poli 245, tak sa tieto názvy zaznamenávajú v poli 246. Informácie v poli 246 prispievajú k ďalšej identifikácii diela a zvyšujú pravdepodobnosť spojenia toho istého diela, napríklad v prípade, keď je dielo vydané v rôznych prekladoch.

Pole	Podpole	poznámka
245	a	Názov
	b	Zvyšok názvu
246	a	Iná forma názvu
765	t	Originálny názov (v prípade prekladu)

Ukážka:

245 1 0 \a Web design \b nenuťte uživatele přemýšlet! \c Steve Krug; [překlad Jan Škvařil]

765 0 \t Don't make me think: common sense approach to Web usability \a Krug, Steve

245 0 0 \a ALA bulletin

246 2 \a American Library Association bulletin

Unifikovaný názov

Unifikovaný názov sa používa vtedy, ak sa dielo objavilo pod meniacimi sa názvami. V tomto prípade sa musí na reprezentáciu diela zvoliť jeden konkrétny názov. Toto pole sa nachádza v bibliografickom zázname len v prípade, ak záznam obsahuje pole 100 (Hlavný vstup - Osobné meno), pole 110 (Hlavný vstup - Korporatívne meno) alebo pole 111 (Hlavný vstup -Meno zhromaždenia). Názov, ktorý sa vyskytuje na katalogizovanom diele, sa dáva do poľa 245 (Názov). Pole 240 sa nepoužíva, ak je prítomné pole 130 (Hlavný vstup-Unifikovaný názov).

Pole	Podpole	poznámka
240	a	Názov
	l	Jazyk diela

Ukážka:

240 1 \a Don't make me think: common sense approach to Web usability \l Česky

240 1 0 \a O monte cinco \l Slovensky

Predmet, Tematické heslo, MDT

Polia 600, 610, 630, 650, 651 a 655 obsahujú heslá a termíny na prístup podľa predmetu. Väčšina týchto polí obsahuje predmetové pridané vstupy alebo prístupové termíny vychádzajúce zo zoznamov a súborov autorít. Pre všetky spomenuté polia platí, že v podpoli \2 sa nachádza zdroj hesla alebo termínu.

V poli 080 sa nachádza číslo Medzinárodného desatinného triedenia (MDT). Pole 080 sa môže opakovať aby sa v ňom dali zaznamenať viaceré čísla MDT pridelené dokumentu.

Pole	Podpole	poznámka
600	a	Osobné meno
610	a	Korporatívne meno
630	a	Meno zhromaždenia
650	a	Unifikovaný názov
651	a	Geografický názov
655	a	Indexačný termín-Žáner/Forma

Ukážka

600 0 7 \a Eliáš, \c prorok, \d 10 stor. pr. Kr. \2 SNKPH

630 0 7 \a Biblia \p S.Z. \p Kráľovská, 1. \2 SNKPH

650 0 7 \a starozákonní proroci \2 SNKPH

655 7 \a brazílske romány \2 SNKPH

Identifikátory

Bibliografický záznam obsahuje viacero informácií, ktoré sa dajú použiť na identifikovanie dokumentu a diela. Medzi základnejšie identifikačné údaje patria ISBN, ISSN, odkazy na predchádzajúce vydania a jazyk. Každý bibliografický záznam má vlastný identifikátor, ktorý mu prideľuje digitálna knižnica alebo iná organizácia.

Pole	Podpole	poznámka
001	a	Evidenčné číslo záznamu
003	a	Zdroj evidenčného čísla
020	a	ISBN
022	a	ISSN
041	a	Kód jazyka
490	a	Edícia
490	v	Zväzok/sekvenčné označenie
773	g	Informácia o vzťahu
773	t	Názov hostiteľského dokumentu (žurnál, zbierka)
773	x	ISSN hostiteľského dokumentu (žurnál, zbierka)
773	z	ISBN hostiteľského dokumentu (žurnál, zbierka)
780	t	Väzba na predchádzajúce vydanie - Názov
780	x	Väzba na predchádzajúce vydanie - ISSN
780	z	Väzba na predchádzajúce vydanie - ISBN
785	t	Väzba na nasledujúce vydanie - Názov
785	x	Väzba na nasledujúce vydanie - ISSN
785	z	Väzba na nasledujúce vydanie - ISBN

1.6 Riešenie nejednoznačnosti pojmov

Rozlišovanie pojmov (prevažne mien autorov, korporácií, vydavateľov, atď.) patrí k najväčším problémom knižníc, ktoré vytvárajú a rozširujú bibliografické záznamy. Ako už bolo spomenuté v predchádzajúcich častiach dokumentu, bibliografický záznam obsahuje vo svojej opisnej časti údaje o autorovi, názvov diela, dátum a miesto vydania, informácie o vydavateľovi, rozsahu, obsahu, jazyku dokumentu a ďalšie identifikačné údaje. Bibliografický záznam môže obsahovať aj rôzne klasifikačné

prvky, ktoré umožňujú tematické zaradenie dokumentu. Tieto údaje nemusia byť vždy postačujúce, aby mohli byť všetky pojmy správne rozlíšené, najmä v prípade, keď sa v záznamoch nachádzajú nejednoznačné mená autorov (autor vystupujúci pod rôznymi menami alebo viacero rôznych autorov s rovnakým menom).

Problém rozlišovania mien autorov môžeme formulovať nasledovne:
majme množinu bibliografický záznamov $B = \{b_1, b_2, ..., b_k\}$, ktoré sa majú zaradiť do existujúcej množiny bibliografických záznamov v knižničnom systéme. Prvok b_i je zoznam atribútov ako meno autora, názov diela, vydavateľské údaje, miesto a čas vydania diela, jeho zaradenie, atď. V tomto prípade vyberieme atribút *meno autora*, prislúchajúci menu jedného unikátneho autora. Každý element *meno autora* r_j je odkazom na autora. Cieľom rozlišovacej metódy je vytvoriť funkciu, ktorá rozdelí množinu odkazov $\{r_1, r_2, ..., r_m\}$ na množinu $\{a_1, a_2, ..., a_n\}$, kde každý prvok a_i obsahuje (všetky a ideálne iba všetky) referencie na jedného autora[25].

Na rozlišovanie mien autorov v bibliografických dátach je možné použiť viacero metód. Medzi dve najrozšírenejšie patria metóda učenie s učiteľom a metóda učenia bez učiteľa.

Metóda učenia bez učiteľa je zameraná prevažne na vytváranie zhlukov (klastrov) autorov na základe podobnosti určitých atribútov medzi jednotlivými autormi, s cieľom zjednotiť záznamy patriace jednému autorovi do rovnakého zhluku.

Metódy učenia s učiteľom vyžadujú pred pripravenú množinu trénovacích dát, na základe ktorých vytvárajú model, ktorý dokáže určiť, či sú dvaja autori jedna a tá istá osoba, alebo dokáže zaradiť autora z nového bibliografického záznamu k existujúcemu autorovi v knižničnom systéme. Vzniknutý model je nasledovne použitý na zvolené bibliografické dáta.

Obidve metódy vyžadujú pevnú množinu dát, ktorá sa počas procesu rozlišovania nesmie meniť. Zaradenie nových dát je častokrát spojené s opätovným vytváraním nových zhlukov a trénovacej množiny. Tieto operácie môžu byť drahé vzhľadom na to, že knižnice obsahujú tisíce bibliografických záznamov.

V októbri 2011 bola publikovaná metóda s názvom INDi (ang. *an Incremental unsupervised Name Disambiguation method*). Metóda INDi nepotrebuje staticky pred pripravené zhluky ani trénovaciu množinu. Metóda INDi má špecifický princíp rozlišovania a priraďovania nových záznamov do knižničného systému. V prípade nejasného určenia pri rozlišovaní autora sa prednostne vytvorí nový záznam (keď daný autor nemá záznam v knižničnom systéme) ako nesprávne priradenie nejasne určeného záznamu existujúcemu autorovi s pravdepodobnosťou chyby. Metóda vytvára zhluky autorov, ktoré sú "čisté" a väčšina záznamov priradených metódou INDi niektorému existujúcemu autorovi sú s najväčšou pravdepodobnosťou správne.

Vedľajším efektom tejto metódy je jav, keď sa autorovi, ktorý nemá veľa záznamov, môže stať, že sa jeho tvorba rozdelí medzi viacero zhlukov, pretože metóda nemala dostatok informácií, aby ich mohla spojiť do jedného zhluku. Hlavným dôvodom takéhoto prístupu je fakt, že manuálna oprava zmiešaných záznamov viacerých autorov je oveľa ťažšia úloha, ako spojiť záznamy toho istého autora dokopy[26].

[25] Incremental Unsupervised Name Disambiguation in Cleaned Digital Libraries, s.289 -290
[26] Incremental Unsupervised Name Disambiguation in Cleaned Digital Libraries, s. 291

Pseudokód INDi algoritmu[27]:

Vstup: bibliografický záznam: c, C je databáza knižnice, prahy podobností αVenue a αT itle, inkrementačná hodnota δ.
Výstup: zoznam L obsahujúci dvojice - referenciu r a autora z databázy knižnice (identifikátor autora).

```
1: c′← PreprocessCitationRecord(c);
2: for each reference r ∈ c′ do
3:    S ← GetClustersOfSimilarAuthors(r, C);
4:    S′← PreprocessClusters(S);
5:    if there is cluster s ∈ S′ and similarCoauthors(s, r) and (similarTitle(s, r, αTitle) or
      similarVenue(s,r, αVenue)) then
6:          add(L,r,s.IdAuthor);
7:    else
8:          auxThresVenue ← αVenue + δ;
9:          auxThresTitle ← αTitle + δ;
10:               if r.coauthorList is empty then
11:                    if there is cluster s ∈ S′ and (similarTitle(s, r, auxThresTitle) or
                       similarVenue(s, r, auxThresVenue)) then
12:                         add(L,r,s.IdAuthor);
13:                   else
14:                         add(L,r,newIdAuthor());
15:              end if
16:              else
17:                   if there is s ∈ S′ and s.coauthorList is empty and (similarTitle(s, r,
                      auxThresTitle) or similarVenue(s, r, auxThresVenue)) then
18:                         add(L,r,s.IdAuthor);
19:                   else
20:                         add(L,r,newIdAuthor());
21:                   end if
22:              end if
23:         end if
24: end for
```

[27] Cit. 1, s. 292

Opis algoritmu:
V prvom kroku algoritmus predspracuje bib. záznam. Z mien a pojmov odstráni interpunkciu. Následne sa spustí cyklus pre každého autora v bib. zázname. Z databázy knižnice sa vyberú všetky záznamy, v ktorých sa nachádza rovnaké alebo podobné meno s menom práve spracovávaného autora. Zo záznamov sa vytvoria logické zhluky, ktoré sa predspracujú ako v kroku 1. V krokoch 5 až 23 je snaha nájsť autora pre danú referenciu *r*. V krokoch 5 a 6 sa hľadá taký zhluk *s*, ktorý obsahuje aspoň jedného spoločného spoluautora a zhodu v názve diela alebo zhodu v publikačných údajoch. V prípade zhody sa referencia *r* priradí autorovi zo zhluku *s*.

Ak sa nepodarilo nájsť zhodu, pokračuje sa krokmi 8 a 9, kde dochádza k úprave prahových hodnôt pre určovanie pomocou podobnosti diela a publikačných údajov, pretože v ďalších krokoch sa už nehľadá zhoda v spoluautoroch.

V ďalších krokoch (10 až 22) dochádza k porovnávaniu a hľadaniu zhody v názve diela alebo publikačných údajov. Algoritmus pracuje so zmenenými prahovými hodnotami. Výsledkom je buď priradenie referencie *r* existujúcemu autorovi z databázy knižnice, alebo vytvorenie nového záznamu pre autora z referencie *r*, pretože sa v databáze s najväčšou pravdepodobnosťou ešte nenachádza.

Algoritmus sa ukončí ako náhle prejde celý zoznam autorov z bibliografického záznamu.
Pri správnom navrhnutí metód similarCoauthors(),similarTitle(),similarVenue() a určení prahových hodnôt podobností môže byť algoritmus použitý aj na rozlišovanie iných entít ako napr.: korporácií, vydavateľov, diel a akcií.

1.7 Podobnosť textových reťazcov

Ako bolo spomenuté v predchádzajúcej kapitole, pri použití INDi algoritmu je potrebné navrhnúť metódy na zistenie podobnosti spoluautorov, diel a vydavateľských údajov. Vo všetkých troch prípadoch je potrebné vziať do úvahy podobnosť mien a názvov ako textových reťazcov. Rôznorodosť mien, nejednotné štandardy v zápise a formáte mien ("R. E. Ellis" vs. "Randy E. Ellis") a diel, rozdielne diakritické znamienka, preklepy a kombinácie týchto faktorov sú príčinou vzniku variantných foriem mien a názvov. Vyhľadávanie iba exaktnej zhody v mene osoby alebo názve diela teda nemusí prinášať najlepšie výsledky.

Riešenie problému prinášajú metriky na porovnávanie reťazcov. Každá metrika bola špecifická pre daný účel a zohľadňovala špecifické vlastnosti reťazcov, ktoré mala rozlišovať.
Na riešenie problému hľadania zhody medzi variantnými formami mien a názvov sa osvedčili metriky na báze podobnosti znakov (Character-based Similarity Metrics)[28]. Do tejto skupiny zaraďujú metriky:
Dice Distance, Levensthein distance, Smith-Waterman distance, Jaro distance, Jaro-Winkler distance a iné.

1.7.1 Metrika Levensthein distance

Podobnosť slov je v tejto metrike definovaná ako počet znakov, ktoré musia byť zmenené na pretransformovanie prvého reťazca na druhý reťazec. Metrika rozširuje metriku Hamming distance[29]. Poskytuje navyše možnosť pridať alebo zmazať znak. Počet znakov v slove nemusí byť rovnaký ako v prípade metriky Hamming distance. Vďaka čomu sa stáva použiteľnú pre všetky reťazce. Nedostatkom metriky je jej neschopnosť rozoznávať, kde nastala zmena. Je totiž rozdiel, či sa zmena nachádza na okraji alebo vo vnútri slova.

[28] BYUNG-WON, ON. Social Network Analysis on Name Disambiguation and More, s. 1081
[29] KUŠŤÁROVÁ, T. Miery podobnosti reťazcov, s. 23

1.7.2 Metrika Jaro distance

Metrika Jaro distance je vzdialenosť, ktorá rozširuje Dice distance, pričom ale zohľadňuje pomery počtu spoločných znakov a dĺžku reťazcov. Metrika berie do úvahy aj pomer spoločných znakov a transpozícií, teda výskyt opakujúcich sa spoločných znakov.
Metrika Jaro distance zobrazuje vzdialenosti porovnávaných slov na interval <0,1>. Hodnota 0 predstavuje rozdielne slová (bez spoločného znaku) a 1 predstavuje identické slová. Počítanie transpozícií je vhodné, ak chceme zistiť, či niektoré znaky v slovách neboli vymenené, napríklad pri kontrole pravopisu, alebo keď potrebujeme zistiť, ktoré slovo chcel používateľ napísať, ale pomýlil sa.

Metrika Jaro distance dokáže nájsť podobnosť slov ako napríklad odysea a odysae, kde ich vzdialenosť bude 0,944. Preto je Jaro distance vhodná na rozpoznávanie pravopisných chýb[30].

1.7.3 Metrika Jaro-Winkler distance

Metrika Jaro-Winkler distance rozširuje metriku Jaro distance. Výpočet sa realizuje podobne, rozdiel je v zohľadnení dĺžky spoločného prefixu. Vďaka tomu poskytuje presnejšie výsledky pre reťazce, ktoré majú spoločný prefix. Metrika je vhodná na meranie vzdialeností slovenských slov, rovnako ako aj slov v jazykoch, v ktorých sa vyskytuje časovanie a skloňovanie, pretože berie do úvahy spoločný prefix slov.

Tabuľka č. 1 Vzdialeností slov vypočítané metrikami Levensthein distance a Jaro-Winkler distance.

Slovo 1	Slovo 2	Levensthein distance	Jaro-Winkler distance
Lóderer	Loderer	1	0.9142858
Lóderer	Lderer	1	0.95714283
Revny, P.	Rovny, P.	1	0.93333334
Marta Lódererová	Milada Lódererová	4	0.8667017
STU	SPU	**1**	**0.8000001**
Ľubomír Kaššák	Lubomir Kassak	5	0.7619047
Slovenská poľnohospodárska univerzita v Nitre	Slovenská poľnohospodárska univerzita	**8**	**0.9762963**
Slovenská poľnohospodárska univerzita	Slovenská technická univerzita	16	0.89720124
Štátny geologický ústav D. Štúra	Štátny geologický ústav Dionýza Štúra	6	0.9613085

[30] KUŠŤÁROVÁ, T. Miery podobnosti reťazcov, s. 23

Porovnávané metriky nevracajú výsledky na rovnakej škále, no aj napriek tomu, je viditeľné, že metrika Jaro-Winkler poskytuje väčšiu presnosť. Navyše nie je obmedzená na celočíselné hodnoty z intervalu <0, počet znakov dlhšieho slova>. Interval reálnych čísiel <0,1> poskytuje väčšiu mieru škálovateľnosti.

1.8 Informácie z externých systémov

V niektorých prípadoch, informácie obsiahnuté v bibliografickom zázname nie sú postačujúce na jednoznačné identifikovanie autora, diela, vydavateľa, korporácie, atď. Preto je potrebné využiť informácie z voľne dostupných zdrojov ako Medzinárodný súbor autorít VIAF alebo GoogleBooks. Obidva zdroje obsahujú vo svojich databázach informácie o knihách vydaných na Slovensku. Aj napriek tomu, že Slovensko v súčasnosti nemá vybudovaný národný súbor autorít, ktorý je nevyhnutnou podmienkou pre spoluprácu s VIAFom, je možné v databáze systému VIAF nájsť autorov publikujúcich na Slovenku, ktorí sa do tejto databázy dostali z iných (prevažne českých) knižničných systémov.

1.8.1 GoogleBooks

GoogleBooks je vyhľadávacia služba, ktorej cieľom je sprístupňovať digitalizované knihy prostredníctvom internetu. V súčasnosti obsahuje databáza GoogleBooks viac ako 15 miliónov naskenovaných kníh.[31]

GoogleBooks poskytuje webovú RESTful službu využívajúcu HTTP a HTTPS protokol na výber a modifikáciu dát v rámci systému Google. Klient špecifikuje operáciu použitím HTTP príkazov POST, GET, PUT alebo DELETE.

1.8.1.1 Vyhľadávanie kníh

Vyhľadávanie sa uskutočňuje zaslaním http GET požiadavky na server. Štandardná podoba URI:

https://www.googleapis.com/books/v1/volumes?q=<hľadaný+výraz>

q – (query) vyhľadávajú sa knihy, ktoré obsahujúce textový reťazec. Ak sa vyhľadáva viac slov, je potrebné spojiť tieto slová znamienkom + (má význam AND). Pri vyhľadávaní sa môžu použiť špeciálne parametre upresňujúce formuláciu dopytu: intitle, inauthor, inpublisher, subject, isbn.[32]
Výsledkom vyhľadávania je JSON súbor, v ktorom sa nachádza počet nájdených kníh a záznamy jednotlivých kníh. JSON súbor môže obsahovať nula, jeden alebo viac záznamov (*volumeresource*).

1.8.1.2 VolumeResource – záznamy kníh

VolumeResoure je záznam (množina informácií), ktoré GoogleBooks uchováva o každej knihe. Informácie sú štruktúrované do niekoľkých kategórií. Výber informácií zo záznamu volumeResorce, ktoré môžu byť použité pri identifikácii diela sa nachádzajú v tabuľke č. 2.

[31] Google code. New Books API fordevelopers [online]
[32] Google code. Books API- Refernece [online]

Tabuľka 1. Výber niektorých atribútov, ktoré poskytuje služba GoogleBooks

Názov atribútu	Typ	Popis
id	string	Jedinečný identifikátor knihy.
selfLink	string	Odkaz URL na túto knihu.
volumeInfo	objekt	Všeobecné informácie o knihe
volumeInfo.title	string	Názov
volumeInfo.subtitle	string	Podtitul
volumeInfo.authors	list	Mená autorov
volumeInfo.publisher	string	Meno vydavateľstva
volumeInfo.publishedDate	string	Dátum publikovania
volumeInfo.description	string	Krátky abstrakt knihy. Text obsahuje značky jazyka HTML.
volumeInfo.industryIdentifiers	list	Štandardné priemyselné označenie knihy
volumeInfo.industryIdentifiers.type	string	Typ identifikátora: ISBN_10, ISBN_13
volumeInfo.industryIdentifiers.identifier	string	Konkrétny identifikátor
volumeInfo.pageCount	integer	Počet strán
volumeInfo.language	string	Jazyk, v ktorom je kniha napísaná. Obsahuje dvojpísmenové označenie podľa normy ISO 639-1. Napr. en, fr, sk.

Zoznam všetkých atribútov je možné nájsť na stránke GoogleBooks.[33]

1.8.2 VIAF

Projekt VIAF - Virtual International Authority File - Medzinárodný virtuálny súbor autorít osobných mien, vytvorený organizáciou OCLC (Online Computer Library Center). Cieľom projektu je vytvoriť automatické prelinkovanie viacerých národných súborov autorít.

VIAF poskytuje jednotný identifikátor pre entity, ktoré sa nachádzajú v národných súboroch autorít, čím prispieva k tvorbe sémantického webu. V súčasnosti VIAF pokrýva osobné mená, mená korporácii, mená rodín, prác a geografické názvy. Vo VIAF registri sa nachádza viac ako 14 miliónov mien. Všetky záznamy jednej entity (z rôznych súborov autorít) spájajú do jedného "super" záznamu, ktorý obsahuje rôzne mená a originálne identifikátory danej entity.

[33] Google code. Books API- Refernece [online]

Systém VIAF poskytuje webové RESTfull API, založené na GET dopytoch. Vyhľadávanie sa uskutočňuje zaslaním http GET požiadavky na server. API poskytuje niekoľko formátov výstupu: MARC21, JSON, RDF alebo VIAF XML. Zoznam poskytovaných informácií sa nachádza v tabuľke č. 3.

Tabuľka č. 3 Informácie a polia obsiahnuté vo VIAF zázname (formát MARC21).

Hodnota	Pole	Podpole
Meno osoby	700	a
Názov korporácie, názov vydavateľstva	710	a
Alternatívne názvy	410	a
Diela	910	a
Počet autor vybraného diela	910	9
ISBN diel	901	a
Mená spoluautorov	950	a
Počet spoločných diel s daným autorom	950	9
Mená spolupracujúcich korporácií	951	a
Počet spoločných diel s danou korporáciou	951	9
Miesta vydania diel (skratka krajiny)	922	a
Počet diel vydaných v danej krajine	922	9
Názvy vydavateľov	921	a
Počet spoločných diel s daným vydavateľom	921	9

1.9 Identifikované problémy

Počas analýzy boli identifikované nasledovné problémy:

- Problém s vyhľadávaním sofistikovaných informácii v oblasti vedy a výskumu v bibliografických dátach - chýbajú sémantické vzťahy, ktoré by uľahčili vyhľadávanie v dátach.
- daný problém môže vyriešiť použitie VIVO ontológie, ktorá obsahuje potrebné bibliografické entity a vzťahy medzi nimi. Bibliografické dáta z knižničných systémov sa transformujú do nového VIVO RDF modelu, v ktorom bude možné pokročilé vyhľadávať pomocou SPARQL dopytov.
- s transformáciou bibliografických dát je nevyhnutne spojené vytvorenie mapovania z MARC formátov na triedy a väzby VIVO ontológie.
- MARC záznamy môžu obsahovať chyby spôsobené ľudským faktorom (preklepy) a nesprávne vyplnené polia a podpolia - cieľom je vytvoriť metódy na automatickú opravu.

- Ďalším objaveným problémom je absencia jednoznačných identifikátorov na národnej úrovni. Na Slovensku nie je vytvorený národný súbor autorít, čo spôsobuje problém nejednoznačnosti mien autorov, najmä pri prepájaní autorov z viacerých knižničných systémov a iných bibliografických databáz. Každý knižničný systém využíva iba vlastné lokálne identifikátory na označenie autorov, diel, korporácií, vydavateľov, atď.
- Na identifikáciu inštancií autorov, diel a vydavateľov sa využije INDi algoritmus, ktorý sa snaží zabrániť chybnej identifikácií dvoch rôznych inštancií. V prípade, keď informácie

z bibliografického záznamu nebudú postačujúce na identifikovanie inštancie, tak sa automaticky využijú informácie z externých systémov VIAF a GoogleBooks.

- Prepojením bibliografických dát so systémom VIAF pomocou identifikátora viafid sa zapájame do projektu Linked Data. Dokážeme tak získavať ďalšie informácie o entitách z externých zdrojov.

Cieľom práce je vytvoriť automatizovaný systém, ktorý dokáže napĺňať vybrané ontológie a na základe vopred určených pravidiel vytvárať sémantické vzťahy, ktoré sa nachádzajú v bibliografických záznamoch. Súčasťou práce a hlavným prínosom bude zlepšenie metód na rozlišovanie inštancii a ich rôznych pomenovaní. Napríklad zjednotiť rôzne mená toho istého autora spôsobené preklepmi pri vytváraní bibliografických záznamov alebo rozlišovať viacerých autorov s rovnakým menom.

2 Špecifikácia

V časti špecifikácia je uvedený opis systému spolu s funkciami, ktoré sa od neho očakávajú. V kapitole 1.2 sa nachádzajú identifikované prípady použitia.

2.1 Opis systému

Systém bude slúžiť na poloautomatické spracovanie bibliografických MARC záznamov z knižničných systémov. MARC záznamy môžu byť vo formáte MARC21 alebo UNIMARC. Hlavnou úlohou systému bude transformácia bibliografických záznamov a ich prevod do RDF tripletov pre zvolenú ontológiu.
Systém bude využívať VIVO ontológiu a príslušné ontológie, s ktorými VIVO spolupracuje. Úplný zoznam použitých ontológií sa nachádza v časti Technická dokumentácia, kapitola E: použité ontológie.

Jednou z kľúčových úloh systému bude správne rozlíšenie mien inštancií (napr.: rozlíšenia rôznych autorov s rovnakými menami alebo rovnakých autorov s preklepmi v mene). Rozlišovanie mien sa týka všetkých entít v systéme: osôb, diel a vydavateľov.

Okrem dát získaných z knižničných systémov sa budú pri rozlišovaní inštancií využívať externé systémy VIAF ako overený medzinárodný súbor autorít a GoogleBooks, ktoré poskytnú doplňujúce informácie, ktoré nemusia byť súčasťou alebo obsiahnuté v bibliografickom zázname.
Systém umožní používateľovi špecifikovať jednotlivé polia a podpolia MARC záznamov, ktoré budú použité pri transformácií. Používateľ bude taktiež rozhodovať o použití externých systémov VIAF a GoogleBooks pri rozlišovaní inštancií.

Systém bude realizovaný ako počítačový program, ktorý poskytne používateľovi potrebné grafické rozhranie na manažovanie a pribežné monitorovanie transformácie dát. Súčasťou programu bude MySQL databáza,

v ktorej budú uložené niektoré konfiguračné nastavenia a dáta v procese predspracovania.
Výstup systému bude zobrazený v grafickom rozhraní VIVO systému, kde je používateľovi umožnené prehliadať, vyhľadávať a modifikovať dáta a väzby medzi spracovanými bibliografickými záznamami.

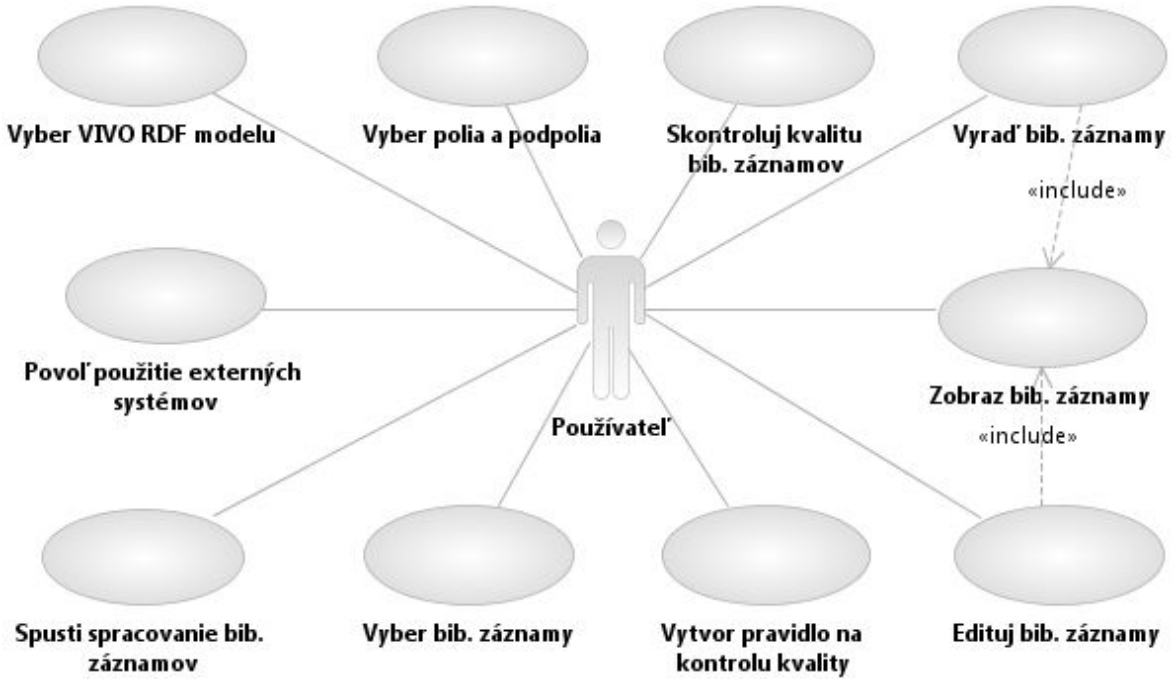
2.1.1 Funkcionálne požiadavky na systém

- výber bibliografických dát na spracovanie
- výber polí a podpolí v MARC záznamoch, ktoré sa použijú v transformácií
- kontrola kvality MARC záznamov
- povolenie alebo zakázanie použiť externé systémy VIAF a GoogleBooks pri rozlišovaní inštancií tried a na obohatenie záznamov o nové alebo chýbajúce prvky
- nastavenie konfiguračných hodnôt systému
- transformácia bib. dát do VIVO ontológie
- nastavenia prahových hodnôt, ktoré vstupujú do algoritmov slúžiacich na rozlišovanie inštancií tried
- prezeranie správ z procesu transformácie
- úprava hodnôt bibliografických údajov v procese predspracovania
- získavanie informácii z portálu GoogleBooks a VIAF
- spracovanie dát z jednotlivých zdrojov
- komunikácia s internou databázou

- ošetrenie chybových stavov
- vytvorenie výstupov vhodných pre zobrazenie v grafickom systéme VIVO

2.2 Prípady použitia

V systéme boli identifikované prípady použitia, ktoré sú zobrazené v diagrame prípadov použitia na obrázku č.5. So systémom pracuje jeden typ používateľa.



Obrázok č. 5. Spojenie prípadov použitia do jedného diagramu

2.2.1 Prípad použitia UC01 Výber polí a podpolí

Vstupné podmienky: Používateľ môže zmeniť prednastavený výber MARC polí a podpolí, z ktorých sa budú čerpať informácie pre jednotlivé entity v ontológii.

Výstupné podmienky:

1. Upravený zoznam MARC polí a podpolí.
2. Pôvodný zoznam

Účastníci: Používateľ

Hlavný tok:

1. Používateľ si zvolí možnosť Upraviť zoznam polí a podpolí.
2. Používateľ si zvolí jednu z možnosti: MARC21 alebo unimarc polia
3. Systém vyhľadá zoznam polí podľa vybraného formátu MARC.
4. Systém zobrazí zoznam polí, z ktorých sa čerpajú informácie spolu s entitami do ktorých sa dané informácie zapisujú

5. Používateľ si prezrie zoznam.
6. Používateľ vykoná zmeny v nastaveniach.
7. Používateľ zvolí uloženie zmien, ktoré vykonal.
8. Systém uloží zmeny.
9. Prípad použitia končí.

Alternatívne toky:

A1: Používateľ nevykoná zmeny

Tok sa aktivuje miesto kroku 6.

1. Používateľ nevykonal žiadne zmeny v nastaveniach.
2. Používateľ zvolí opustenie nastavení.
3. Prípad použitia končí.

A2: Používateľ neuloží zmeny

Tok sa aktivuje miesto kroku 8.

1. Používateľ zvolí opustenie systému bez uloženia zmien.
2. Používateľ zvolí opustenie nastavení.
3. Systém neuloží zmeny.
4. Prípad použitia končí.

2.2.2 Prípad použitia UC02 Výber bibliografických záznamov

Vstupné podmienky: Používateľ zvolí bibliografické záznamy (xml súbor), ktoré systém zaradí do procesu predspracovania.

Výstupné podmienky:

1. Zaradenia vybraných záznamov do procesu predspracovania.
2. Žiadne záznamy nie sú vybrané.

Účastníci: Používateľ

Hlavný tok:

1. Používateľ si zvolí možnosť Vybrať dáta/súbor na spracovanie.
2. Systém ponúkne možnosť vybrať súbor na spracovanie.
3. Používateľ vyberie súbor.
4. Používateľ zvolí formát záznamov (MARC21 alebo UNIMARC)
5. Používateľ spustí nahrávanie záznamov do databázy
6. Systém uloží zvolené záznamy do databázy.
7. Systém na základe vopred vybraných polí a podpolí pre zvolený MARC formát vyextrahuje informácie z jednotlivých bib. záznamov a uloží ich do databázy.
8. Používateľ si prezrie štatistické údaje.
9. Prípad použitia končí.

Alternatívne toky:

A1: Používateľ nezvolí žiadny súbor

Tok sa aktivuje miesto kroku 3.

1. Používateľ nezvolí žiadny súbor.
2. Používateľ zvolí ponuku opustiť výber súboru.
3. Prípad použitia končí.

A2: Používateľ nezvolí nahrávanie súborov do databázy

Tok sa aktivuje miesto kroku 5.

1. Používateľ nezvolí nahrávanie súborov do databázy.
2. Používateľ zvolí ponuku opustiť výber bib. záznamov.
3. Prípad použitia končí.

2.2.3 Prípad použitia UC03 Kontrola kvality vybraných bib. záznamov

Vstupné podmienky: Používateľ zvolí možnosť skontrolovať kvalitu bib. záznamov, na základe vopred definovaných kontrolných pravidiel.

Výstupné podmienky:

1. Zoznam záznamov, ktoré spĺňajú definované kontrolné pravidlá.
2. Zoznam záznamov, ktoré nespĺňajú definované kontrolné pravidlá.

Účastníci: Používateľ

Hlavný tok:

1. Používateľ si zvolí možnosť Kontrola bib. záznamov, ktoré boli zaradené do procesu predspracovania.
2. Systém zobrazí zoznam existujúcich pravidiel kontroly kvality.
3. Používateľ vyberie zo zoznamu kontrolné pravidlá, ktoré chce aplikovať na bib. záznamy a spustí kontrolu.
4. Systém vykoná kontrolu záznamov.
5. Používateľ si prezrie výsledky kontroly kvality záznamov.
6. Prípad použitia končí.

Alternatívne toky:

A1: Používateľ si nevyberie zo zoznamu pravidiel kontroly

Tok sa aktivuje miesto kroku 3.

1. Používateľ nezvolí žiadne z existujúcich pravidiel kontroly.
2. Používateľ zvolí ponuku opustiť kontrolu bib. záznamov.
3. Prípad použitia končí.

2.2.4 Prípad použitia UC04 Vytvorenie nového pravidla kontroly kvality

Vstupné podmienky: Používateľ zvolí možnosť Vytvoriť nové pravidlo kontroly kvality bib. záznamov.

Výstupné podmienky:

1. Nové pravidlo kontroly, ktoré je zaradené do zoznamu existujúcich pravidiel.
2. Nevytvorenie nového pravidla.

Účastníci: Používateľ

Hlavný tok:

1. Používateľ zvolí možnosť na vytvorenie nového kontrolného pravidla.
2. Systém ponúkne možnosť výberu MARC formátu, na ktorý sa bude pravidlo aplikovať.
3. Používateľ zvolí MARC formát, pole a podpole prípadne polia a podpolia.
4. Používateľ zadá názov pravidla, kontrolné pravidlo a v prípade potreby zadá požadované údaje (napr.: číselník alebo iné konštanty v závislosti od typu pravidla).
5. Používateľ zvolí možnosť uložiť pravidlo.
6. Systém uloží pravidlo a do ponuky existujúcich pravidiel.
7. Prípad použitia končí.

A1: Používateľ nevyberie zo zoznamu pravidiel kontroly

Tok sa aktivuje miesto kroku 5.

1. Používateľ nezvolí uloženie nového pravidla.
2. Používateľ zvolí ponuku opustiť tvorbu nového pravidla.
3. Prípad použitia končí.

2.2.5 Prípad použitia UC05 Výber VIVO modelu

Vstupné podmienky: Používateľ zvolí RDF VIVO model, do ktorého sa zapíšu spracované bibliografické záznamy.

Výstupné podmienky:

1. Záznamy sa zapíšu do existujúceho RDF VIVO modelu.
2. Záznamy sa zapíšu do nového RDF VIVO modelu.

Účastníci: Používateľ

Hlavný tok:

1. Používateľ si zvolí možnosť Vybrať RDF model na zápis bib. záznamov, ktoré boli zaradené do procesu predspracovania
2. Systém ponúkne možnosť vybrať existujúci súbor alebo vytvorenie nového súboru.
3. Používateľ vyberie existujúci súbor.
4. Systém overí, čí je zadaný súbor v xml formáte.
5. Systém uloží používateľovu voľbu do nastavení
6. Prípad použitia končí.

Alternatívne toky:

A1: Používateľ vyberie vytvorenie nového súboru

Tok sa aktivuje miesto kroku 3.

1. Používateľ zvolí vytvorenie nového súboru.
2. Systém vytvorí nový RDF VIVO model.
3. Prípad použitia pokračuje krokom 5 v hlavnom toku.

A2: Zadaný súbor nie je platný

Tok sa aktivuje miesto kroku 5.

1. Zvolený súbor nezodpovedá RDF VIVO model formátu.
2. Systém zobrazí chybovú hlášku.
3. Prípad použitia končí.

2.2.6 Prípad použitia UC06 Prezeranie bib. záznamov v procese predspracovania

Vstupné podmienky: Používateľ môže prezerať bib. záznamy, ktoré boli zaradené do procesu predspracovania.

Výstupné podmienky:

1. Zobrazený zoznam bib. záznamov v procese predspracovania.

Účastníci: Používateľ

Hlavný tok:

1. Používateľ si zvolí možnosť prezerať bib. záznamy, ktoré boli zaradené do procesu predspracovania
2. Systém zobrazí zoznam bib. záznamov.
3. Používateľ si prezerá zoznam.
4. Používateľ zvolí ukončenie prezerania bib. záznamov
5. Prípad použitia končí.

2.2.7 Prípad použitia UC07 Editácia bibliografických záznamov

Vstupné podmienky: Používateľ môže editovať bib. záznamy, ktoré boli zaradené do procesu predspracovania.

Výstupné podmienky:

1. Zmenené bib. záznamy v procese predspracovania.
2. Nezmenené bib. záznamy.

Účastníci: Používateľ

Hlavný tok:

1. Používateľ si zvolí možnosť prezerať bib. záznamy, ktoré boli zaradené do procesu predspracovania
2. Systém zobrazí zoznam bib. záznamov.
3. Používateľ si prezerá zoznam.
4. Používateľ si vyberie záznam, ktorý chce upraviť.
5. Systém ponúkne používateľovi možnosť editovať vybraný záznam.
6. Systém zobrazí záznam v editovateľnom rozhraní spolu s možnosťou uložiť zmeny, alebo opustiť bez uloženia
7. Používateľ vykoná zmeny v zázname.
8. Používateľ zvolí možnosť uložiť zmeny.
9. Systém uloží zmenené hodnoty do databázy.
10. Používateľ zvolí ukončenie prezerania bib. záznamov
11. Prípad použitia končí.

Alternatívne toky:

A1: Používateľ si nevyberie žiadny súbor na editovanie

Tok sa aktivuje miesto kroku 4.

1. Používateľ si zvolí ukončenie prezerania bib. záznamov.
2. Prípad použitia končí.

A2: Používateľ neuloží zmeny, ktoré vykonal

Tok sa aktivuje miesto kroku 8.

1. Používateľ si zvolí možnosť ukončiť bez uloženia zmien.
2. Prípad použitia pokračuje krokom 2 v hlavnom toku.

A3: Používateľ chce editovať ďalšie záznamy

Tok sa aktivuje miesto kroku 10.

1. Prípad použitia pokračuje krokom 2 v hlavnom toku.

2.2.8 Prípad použitia UC08 Vyradenie bib. záznamu zo spracovania

Vstupné podmienky: Používateľ môže vylúčiť bib. záznamy, ktoré boli zaradené do procesu predspracovania.

Výstupné podmienky:

1. Zmenené bib. záznamy v procese predspracovania.
2. Nezmenené bib. záznamy.

Účastníci: Používateľ

Hlavný tok:

1. Používateľ si zvolí možnosť prezerať bib. záznamy, ktoré boli zaradené do procesu predspracovania
2. Systém zobrazí zoznam bib. záznamov zaradených do predspracovania.
3. Používateľ si prezerá zoznam.
4. Používateľ si vyberie záznam, ktorý chce vylúčiť zo spracovania.
5. Používateľ potvrdí vylúčenie záznamu.
6. Systém odstráni vybraný záznam z procesu predspracovania.
7. Používateľ zvolí ukončenie prezerania bib. záznamov
8. Prípad použitia končí.

Alternatívne toky:

A1: Používateľ si nevyberie žiadny súbor na vylúčenie

Tok sa aktivuje miesto kroku 4.

1. Používateľ si zvolí ukončenie prezerania bib. záznamov.
2. Prípad použitia končí.

A2: Používateľ chce vylúčiť ďalšie záznamy

Tok sa aktivuje miesto kroku 7.

1. Prípad použitia pokračuje krokom 2 v hlavnom toku.

2.2.9 Prípad použitia UC09 Povolenie použitia externých systémov

Vstupné podmienky: Používateľ môže povoliť alebo zakázať použitie externých systémov pri rozlišovaní a obohacovaní spracovávaných bib. záznamov.

Výstupné podmienky:

1. Jeden alebo viaceré externé systémy sú povolené a zaradené do procesu spracovania.
2. Externé systémy nie sú povolené.

Účastníci: Používateľ

Hlavný tok:

1. Používateľ si zvolí možnosť zobraziť externé systémy, ktoré budú zaradené do procesu predspracovania
2. Systém zobrazí zoznam externých systémov a indikátor, vyjadrujúci zaradenie, alebo nezaradenie do procesu spracovania.
3. Používateľ si prezerá zoznam externých systémov.
4. Používateľ urobí zmeny v nastaveniach.
5. Systém uloží zmenené hodnoty.
6. Prípad použitia končí.

Alternatívne toky:

A1: Používateľ nevykoná zmeny v nastaveniach externých systémov

Tok sa aktivuje miesto kroku 4.

1. Používateľ si nevykoná žiadne zmeny v nastaveniach.
2. Prípad použitia končí.

2.2.10 Prípad použitia UC10 Spustenie spracovania bib. záznamov

Vstupné podmienky: Používateľ spustí spracovanie bib. záznamov, ktoré prešli procesom predspracovania a nachádzajú sa v databáze systému.

Výstupné podmienky:

1. Spracované záznamy v podobe RDF VIVO modelu.

Účastníci: Používateľ

Hlavný tok:

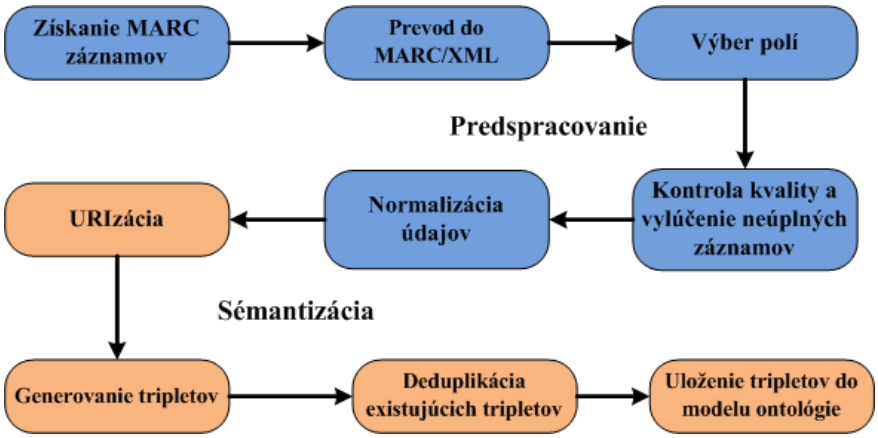
1. Používateľ si zvolí možnosť Spustiť spracovanie.
2. Systém načíta konfiguračné nastavenia.
3. Systém spustí spracovanie pred pripravených záznamov nachádzajúcich sa v databáze systému.
4. Systém ukončí spracovanie a vytvorí výstupný RDF VIVO model.
5. Systém zobrazí používateľovi prehľad o spracovaných záznamoch.
6. Používateľ si prezrie prehľad zo spracovania.
7. Prípad použitia končí.

3 Návrh

V časti návrh je uvedený opis transformácie/sémantizácie bibliografických dát s postupmi na identifikáciu inštancií autorov, vydavateľov a diel. V kapitole sa nachádzajú aj biznis procesy, ktoré bližšie určujú tok údajov a spôsob identifikácie jednotlivých inštancií v procese transformácie. V závere kapitoly sa nachádza návrh architektúry systému a logický dátový model.

3.1 Návrh transformácie bibliografických záznamov

Proces transformácie môžeme logicky rozdeliť na dve hlavné časti. Prvou časťou je predspracovanie bibliografických údajov a druhou časťou je následná sémantizácia a prevod dát do zvolenej ontológie. Celý proces je detailnejšie znázornený na obrázku č. 6.



Obrázok č. 6 Navrhovaný proces transformácie

3.2 Predspracovanie

Proces predspracovania sa skladá z viacerých postupných krokov zameraných na získanie bib. dát, ich kontrolu a úpravu.

3.2.1 Získanie MARC záznamov

Záznamy je možné získať(importovať) viacerými spôsobmi, v závislosti od nastavenia systému a požiadaviek používateľa.

1. **Získanie dát pomocou protokolu OAI PMH**

   Komunikácia je možná iba s knižnično-informačnými systémami, ktoré dokážu spracovať dotazy OAI-PMH. Komunikácia medzi naším a knižničným systémom, je realizovaná pomocou protokolu HTTP, presnejšie zasielaním GET alebo POST požiadaviek. Každý dotaz sa skladá z URL adresy repozitára a zoznamu argumentov vo forme kľúč = hodnota oddelených znakom &. Kľúčom je slovo verb a hodnotou buď Identify, GetRecord, ListIdentifiers, ListMetadataFormats, ListRecords alebo ListSets[34].

[34] BELLA, M. Implementace OAI-PMH pro český WebArchiv, s. 10

Hodnoty ostatných dvojíc závisia na zvolenom príkaze. Vrátený XML dokument obsahuje odpoveď na daný dotaz alebo chybové hlásenie.

2. Získanie dát pomocou protokolu Z39.50

Protokol Z39.50 sa využíva najmä v prostredí knižníc a iných informačných inštitúciách. Prostredníctvom Z-relácie a definovaných funkcií a služieb[35] sa vytvorí medzi klientom (v tomto prípade náš systém) a serverom (zvolený knižničný informačný systém) spojenie, počas ktorého nastáva vzájomná výmena informácií. Formuláciou správnych dopytov v jazyku Z, je možné získať požadované bib. záznamy a príslušné metadáta[36].

3. Import MARC/XML záznamov

Záznamy sú vyexportované z knižničných systémov vo forme MARC/XML súborov, ktoré sa jednoducho nahrajú do systému. Týmto spôsobom je možné veľmi jednoducho preniesť celé korpusy bib. záznamov z knižnično-informačných systémov aj v prípade, keď náš, alebo externý systém nemá internetové pripojenie, ktoré sa vyžaduje v predchádzajúcich dvoch spôsoboch.

3.2.2 Prevod MARC formátu do MARC/XML

Mnohé knižnično-informačné systémy dokážu exportovať bib. záznamy vo forme MARC/XML. V opačnom prípade je potrebné použiť programové knižnice, ktoré vykonajú prevod z MARC formátu do ekvivalentnej podoby MARC/XML. V súčasnosti existuje viacero programových knižníc určených pre rôzne programovacie jazyky ako napr.: MARC4J[37], CSharp MARC[38] a iné, ktoré umožňujú implementáciu vlastného prevodového mechanizmu.

3.2.3 Výber polí a podpolí

Obsah polí a podpolí závisí od použitých katalogizačných pravidiel a MARC formátu. Nie všetky polia a podpolia resp. údaje z nich musia byť transformované do VIVO modelu. Môže ísť o interné identifikátory jednotlivých knižnično-informačných systémov, ktoré by vo vytváranom VIVO modeli nemali žiadnu alebo len minimálnu prídavnú informačnú hodnotu. Výber polí a podpolí spolu s ich mapovaním na jednotlivé triedy, vlastnosti a vzťahy, je uvedený v technickej dokumentácii v časti F- Prevodová tabuľka. Všetky údaje získané z vybraných polí sú ukladané do databázy.

3.2.4 Kontrola kvality a vylúčenie neúplných záznamov

Systém automaticky vylúči, resp. označí záznamy, ktoré nezodpovedajú potrebným kritériám:

1. Chýbajúce polia a podpolia

Každý MARC formát ma vlastnú množinu povinných polí a podpolí, ktoré musí obsahovať. Systém bude kontrolovať prítomnosť definovaných povinných tagov a podpolí v záznamoch. Zoznamy

[35] Ariadne. Z39.50 for All [online]
[36] MOORE, J. The Z39.50 information retrieval standard, s. 145
[37] https://github.com/marc4j/marc4j
[38] http://www.csharpmarc.net/

povinných tagov a podpolí budú uložené v databáze systému. Používateľ bude môcť tieto zoznamy upravovať podľa vlastnej potreby alebo v prípade, keď dôjde k zmenám v katalogizačných pravidlách.

2. Hodnoty v podpoliach

Niektoré podpolia v MARC záznamoch môžu obsahovať len hodnoty z vopred stanovených číselníkov. Typickým číselníkom je napríklad jazyk diela (slo, eng, cze,..) alebo rola osoby.

Rola osoby je vo formáte MARC21 zapísaná v tagu 100a v podpoli 4 alebo aj v tagu700 a tu tiež v podpoli 4. Obsah tohto podpoľa môže nadobúdať hodnoty, ktoré sú priamo definované formátom MARC, ako napríklad:

Vo formáte MARC21 to môžu byť aut - author(autor), edt - editor, trn - translator (prekladateľ) a iné. V UNIMARC-u sú tieto hodnoty vyjadrené zasa číselným kódom, napr.: 005, 430, 740. Sémantická informácie je rovnaká. Preto bude náš systém obsahovať prevodovú tabuľku medzi týmito hodnotami, aby sme mohli reprezentovať túto hodnotu s rovnakým významom.

Zoznamy hodnôt rolí používané pri určovaní zodpovednosti za dielo, ako aj zoznam možných hodnôt určujúcich jazyk dokumentu, a aj ostatné hodnoty, ktoré definuje priamo MARC formát, budú štandardnou súčasťou systému. Ostatné číselníky si môže zadefinovať používateľ na základe štruktúry transformovaných údajov pri konfigurácii systému.

Ako bolo už naznačené, záznamy budú rozdelené na tri základné skupiny. Toto označenie bude stavu záznamu bude slúžiť pri riadení procesu ďalšieho spracovania. Záznamy, ktoré nesplnia kritériá určené pre kontrolu kvality záznamov, budú v databáze označené príznakom s hodnotou "404". Záznamy, ktoré splnia všetky kritériá dostanú príznak "200". Záznamy, ktoré neboli kontrolované majú prednastavenú hodnotu "100". Tieto hodnoty budú zapísané do stĺpca status, v tabuľke marc_record. Fyzický model databázy je uvedený v kapitole Implementácia v časti 4.3.1

Používateľ si môže v tomto kroku prezerať záznamy pripravené na transformáciu. V prípade potreby ich môže upravovať a/alebo vyradiť. Používateľovi bude umožnená aj ručná oprava vylúčených záznamov, ktoré budú po splnení stanovených podmienok opäť zaradené do spracovania.

3.2.5 Normalizácia údajov

V tomto kroku systém aplikuje na dáta normalizačné pravidlá:

- normalizácia dátumov
  extrakcia rokov, mien a znační
- normalizácia ISBN kódov
  vynechávanie pomlčiek a medzier a iného opisného textu
  `978-80-552-0213-6 (brož.)` ➢ `9788055202136`
- úprava zlúčených hodnôt a konštánt
  počet strán: `123 s., 60cm` ➢ `123`
  rozsah strán: `S. 158 - 369` ➢ prvá strana 158, posledná strana 369
  vydanie: `2. vyd.` ➢ `2`

- vylúčenie definovaných konštánt
  napr.: odstránenie konštanty "`[s.l.]`" ak nie je uvedený názov vydavateľa
  odstránenie konštanty "`[s.n.]`" ak nie je uvedené miesto publikovania
- prevod rol osôb z UNIMARC formátu na MARC 21 (napr.: 005 = aut, 430 = edt)
- normalizácia textových informácií (mená a názvy)
  prevod na malé písmená, vynechanie interpunkcie, náhrada ypsilonu za jotu
  `Štefan Hrušovský` ➢ `stefan hrusovski`

Cieľom normalizácie je úprava dát na jednotný formát, ktorý uľahčí ich porovnávanie v nasledujúcich častiach procesu. Normalizované dáta taktiež zvyšujú presnosť pri identifikácií inštancií.

3.3 Sémantizácia

Proces sémantizácie sa je spustený používateľom, po výbere príslušného VIVO modelu a schválení zoznamu predspracovaných dát. Proces sémantizácie sa skladá z viacerých krokov, kde sa medzi najdôležitejšie zaraďuje URIzácia a generovanie tripletov.

3.3.1 URIzácia

Systém sa v tomto kroku vyhľadáva existujúci lokálny URI identifikátor pre vybranú inštanciu v existujúcom VIVO modeli alebo vytvára nový URI identifikátor, ak sa inštancia v modeli ešte nenachádza. V tomto kroku sa vyhľadávajú aj URI identifikátory v externých systémoch VIAF a GoogleBooks. V tejto časti procesu dochádza k identifikácii a rozlišovaniu inštancií tried s rovnakými alebo podobnými názvami, napr.: rozoznávanie autorov s rovnakým menom alebo autorov, ktorí majú preklep v mene. Problémom sú aj variantné formy pomenovania (napr.: SPU, Slovenská poľnohospodárska univerzita). Proces rozoznávania inštancií tried je uvedený v samostatnej kapitole 3.4. Návrh procesu identifikácie inštancií tried.

Navrhnuté URI identifikátory v rámci VIVO modelu budú mať formu:

http://<menný priestor>/<objekt><poradové číslo>
http://<menný priestor>/individual/<objekt><poradové číslo>

Poradové číslo musí byť minimálne 7 ciferné a číslovanie sa začína hodnotou 1000000. Každý objekt má svoje vlastné poradové číslo. Najvyššia hodnota poradového čísle je zapísaná v databáze, v tabuľke idcounter.

V systéme sa budú generovať URI identifikátory pre osoby, diela, vydavateľov a vzťah medzi osobou a dielom.

Ukážka URI:

pre osobu	http://ml.fiit.stuba.sk/individual/person1018117
pre dielo	http://ml.fiit.stuba.sk/individual/book1019280
pre vydavateľa	http://ml.fiit.stuba.sk/individual/publisher1000656
pre vzťah	http://ml.fiit.stuba.sk/relationship1032327

URI identifikátory osôb, diel a vydavateľov musia povinne obsahovať príponu individual/, ktorá sa uvádza za menným priestorom. Príponu individual/ vyžaduje implementácia VIVO systému.

3.3.2 Generovanie tripletov

Po získaní potrebných URI identifikátorov pre všetky inštancie, dochádza k vytváraniu RDF tripletov. Vytvárajú sa vopred definované vzťahy medzi identifikovanými inštanciami a priraďujú sa im vlastnosti získané zo spracovaného bib. záznamu. Na tvorbu tripletov používame VIVO ontológiu, vrátane všetkých príslušných ontológií, na ktoré sa VIVO odkazuje. Zoznam použitých ontológií sa nachádza v technickej dokumentácií v časti E - Použité ontológie. Model vybraných tried VIVO ontológie je uvedený na obrázku č. 3. Navrhnutý zoznam prevodov údajov z tagov a podpolí MARC21 a UNIMARCu do VIVO ontológie uvádzame v technickej prílohe v časti F- Prevodová tabuľka.

3.3.3 Deduplikácia existujúcich tripletov

Z dôvodu minimalizácie modelu a údržby jeho konzistentnosti do modelu ukladáme iba unikátne triplety. Mnohé knižnice pracujúce s RDF modelmi, ako napr. JENA, umožňujú vykonávať deduplikáciu tripletov pri ich zápise do modelu.

3.3.4 Uloženie tripletov do modelu ontológie

Deduplikované triplety, ktoré vznikli spracovaním bib. záznamov, ukladáme v poslednom kroku do vybraného VIVO modelu. Vzniknutý model môžeme importovať do webového rozhrania VIVO systému, ktorý obsahuje nástroje na jeho prezeranie v grafickom rozhraní. Rozhranie ponúka aj interný SPARQL editor na vykonávanie dotazov.

3.4 Návrh procesu identifikácie inštancií tried

V bib. záznamoch sme sa rozhodli identifikovať tri hlavné triedy: dielo, osobu a vydavateľa. V tejto kapitole sú opísané metódy, ktoré navrhujeme použiť pre identifikovanie inštancií spomenutých tried. Každá trieda má vlastné atribúty a vzájomné vzťahy s inými triedami, preto sa navrhované procesy odlišujú. Snahou týchto procesov je čo možno najpresnejšie určenie a nájdenie URI identifikátora v existujúcom VIVO modeli. Ak rozhodovací algoritmus nenájde zhodu medzi práve identifikovanou inštanciou a niektorou inštanciou vo VIVO modeli, tak priradí danej inštancií nový URI identifikátor.
Každý porovnávaný atribút má v procese identifikácie inú váhu a prioritu, preto atribútom priradíme rôzne bodové ohodnotenia, čo poskytne systému škálovateľnosť.

Ako bolo spomenuté v analýze, práca nadväzuje na existujúci INDi algoritmus, ktorý rieši rozoznávanie inštancií tried pomocou porovnávania zvolených atribútov a ich bodovým ohodnocovaním. V rámci návrhu sme definovali atribúty, ktoré sa použijú v INDi algoritme pri rozlišovaní jednotlivých tried.

Porovnávanie každej inštancie triedy (osoba, vydavateľ, dielo) sa vykonáva rovnako a proces môžeme zapísať nasledovným spôsobom:
inštancia vybranej triedy z bib. záznamu sa porovnáva s vybranou množinou inštancií rovnakej triedy z VIVO modelu a vypočítava sa skóre podobnosti pre každú inštanciu z VIVO modelu.

$$\text{Score}_{INŠTANCIA_i} = \sum_{j=1}^{m} Parameter_j * Koeficient_j \quad , \quad m \text{ je počet porovnávaných parameterov} \tag{1.4.a}$$

Pre každú triedu existuje množina parametrov s priradenou kladnou bodovou hodnotou. Zoznam parametrov je uvedený v nasledujúcej kapitole.
Parameter môže nadobúdať iba jednu z dvoch hodnôt:

$$Parameter = \begin{cases} 0 & neexistuje\ zhoda \\ 1 & existuje\ zhoda \end{cases} \tag{1.4.b}$$

Po výpočte skóre sa vyberie inštancia, ktorá nadobudla najvyššiu hodnotu podobnosti:

$$\text{Score}_{\text{INŠTANCIA}_{\text{final}}} = MAX\left(Score_{INŠTANCIA_i}\right),\ i = 1, \dots, n \quad \text{kde } n \text{ je počet porovnávaných inštancií} \tag{1.4.c}$$

Hodnota inštancie s najvyšším dosiahnutým bodovým hodnotením $\text{Score}_{\text{INŠTANCIA}_{\text{final}}}$ sa porovná so stanovenou prahovou hranicou podobnosti. Ak je dosiahnuté skóre vyššie ako prahová hranica, tak sa porovnávaná inštancia z bib. záznamu už nachádza vo VIVO modeli a je jej pridelený URI identifikátor existujúcej inštancie z modelu $URI_{INŠTANCIA\_final}$. V opačnom prípade sa jedná o úplne novú inštanciu, pre ktorú je vygenerovaný nový URI identifikátor.

$$\text{Score}_{\text{INŠTANCIA}_{\text{final}}} = \begin{cases} \geq hranica\_podobnosti \Rightarrow \text{Inštancia sa už nachádza v modeli, } \text{URI}_{\text{INŠTANCIA\_new}} \leftarrow URI_{INŠTANCIA\_final} \\ < hranica\_podobnosti \Rightarrow \text{Inštancia sa ešte nenachádza v modeli, } \text{URI}_{\text{INŠTANCIA\_new}} \leftarrow NEW\ URI \end{cases} \tag{1.4.d}$$

3.4.1 Proces identifikácie autorov

Návrh procesu identifikovania inštancie autora pozostáva zo šiestich krokov.

1. **vytvorenie záznamu o autorovi**

Vytvoríme štruktúru, ktorá bude obsahovať informácie o vybranom autorovi (nazvime ho A) z práve spracovaného bib. záznamu.

Záznam o autorovi A sa skladá z častí:

autor — meno autora a jeho identifikátor v rámci knižničného systému, z ktorého bol záznam získaný(roky, ID , rola).

spoluautori — ostatné osoby, ktoré nesú zodpovednosť za vytvorenie diela (ich mená a identifikátory).

dielo — názov diela a ostatné doplňujúce informácie (paralelné názvy, podtituly, ISBN, ISSN, miesto a rok vydania).

2. **vytvorenie množiny z VIVO modelu pre porovnanie so záznamom o autorovi**

Po vytvorení záznamu o autorovi, je potrebné vytvoriť podobné záznamy, ktoré čerpajú informácie zo zvoleného VIVO modelu. Z modelu sa postupne vyberajú všetky inštancie autorov, ktoré majú rovnaké alebo podobné meno ako autor z kroku 1. V modeli sú uchovávané aj variantné mená autorov. V modeli sa sémanticky vyhľadajú všetky diela daného autora a všetci spoluautori, s ktorými publikoval. Nájdené záznamy vytvoria zhluk.

Každý VIVO záznam sa skladá z častí:

autor — obsahuje meno autora, jeho iné mená, ktoré mu boli priradené vo VIVO modeli, URI identifikátor, identifikátory pod ktorými, daný autor vystupuje v iných externých systémoch a príslušné roky spojené s autorom.

spoluautori — spoluautori daného autora vo VIVO modeli. Spoluautori obsahujú rovnaké informácie ako trieda autor.

diela — názvy a identifikačné údaje(ISBN, ISSN, ID z externých systémov), rok a miesto vydania všetkých diel, ktoré daný autor publikoval a nachádzajú sa vo VIVO modeli.

3. **proces porovnania (INDI algoritmus)**

Autora z bib. záznamu porovnáme s autormi z VIVO modelu, ktorých sme vybrali v kroku 2.
Poradie porovnávaných prvkov je nasledovné:

Priorita	Názov parametru
1.	Zhoda v priezvisku a krstnom mene*
2.	Zhoda v priezvisku a krstnom mene (NT)*
3.	Zhoda v priezvisku a iniciále mena*
4.	Zhoda v priezvisku a iniciále mena (NT)*
5.	Zhoda v roku narodenia
6.	Zhoda v roku úmrtia
7.	Zhoda v identifikátore z iného systému
8.	Zhoda v dielach**
9.	Zhoda v spoluautorstve***

Poznámka:

* Porovnanie nastáva na základe úplnosti mena. Medzi pravidlami 1,2 a 3,4 je vzťah ALEBO. Ak má práve identifikovaná osoba v bib. zázname uvedené celé meno a priezvisko, použijeme atribúty 1a 2. Ak má uvedené iba priezvisko a iniciálu krstného mena, použijeme atribúty 3 a 4.

** Porovnávame na základe primárnych a paralelných názvov ako aj ISBN identifikátorov.

*** Spoluautorov porovnávame na základe identifikátorov, následne podľa mena a priezviska a ak nebola objavená zhoda, nastáva porovnávanie s použitím JWD.

Pri nájdení zhody sa zvyšuje bodové ohodnotenie daného VIVO záznamu. Každý porovnávaný parameter môže mať rozdielnu váhu (koeficient), ktorá sa počíta do celkového ohodnotenia.

Pre každého vybraného autora z VIVO modelu sa vypočíta skóre podobnosti podľa bodovacej funkcie:

$$\text{Score}_{AUTOR} = \sum_{j=1}^{m} Parameter_j * Koeficient_j \quad , \qquad m \text{ je počet porovnávaných parameterov}$$

4. vyhodnotenie a priradenie URI identifikátora

Na základe bodovacej funkcie vyberieme autora z VIVO modelu, ktorý dosiahol najväčšiu podobnosť s autorom A. Môžeme teda tvrdiť, že autor A je s vysokou pravdepodobnosťou tá istá osoba, ako existujúci autor z VIVO modelu.

Toto tvrdenie nie je vždy pravdivé. V niektorých prípadoch je najvyššia dosiahnutá podobnosť medzi autormi malá a nedá sa jednoznačne preukázať, že ide o rovnakú osobu. Napr.: nastala zhoda iba v mene a priezvisku.

Preto je do vyhodnotenia zaradená minimálna bodová hranica podobnosti, ktorú musí autor A a vybraný autor z VIVO modelu dosiahnuť. Ak autor A dosiahol požadovanú hranicu podobnosti, tak mu môžeme priradiť URI identifikátor autora z VIVO modelu. Inak je autorovi A priradený nový URI identifikátor (viď. vzorec 1.4.d).

Ako bolo uvedené v analýze, INDi algoritmus upravuje váhu vybraných parametrov, v prípade, keď do porovnávania nevstupujú žiadni spoluautori. Rozhodli sme sa modifikovať tento prístup. Namiesto úpravy váh všetkých parametrov sme upravili iba jednu hodnotu, ktorou je prahová hranica totožnosti. Ak do porovnania nevstupujú žiadni spoluautori, tak použijeme druhú definovaní prahová hranicu totožnosti (označenú ako Prahová hranica totožnosti bez spoluautorov), ktorá by mala byť spravidla nižšia.

5. použitie externého systému VIAF v procese identifikácie

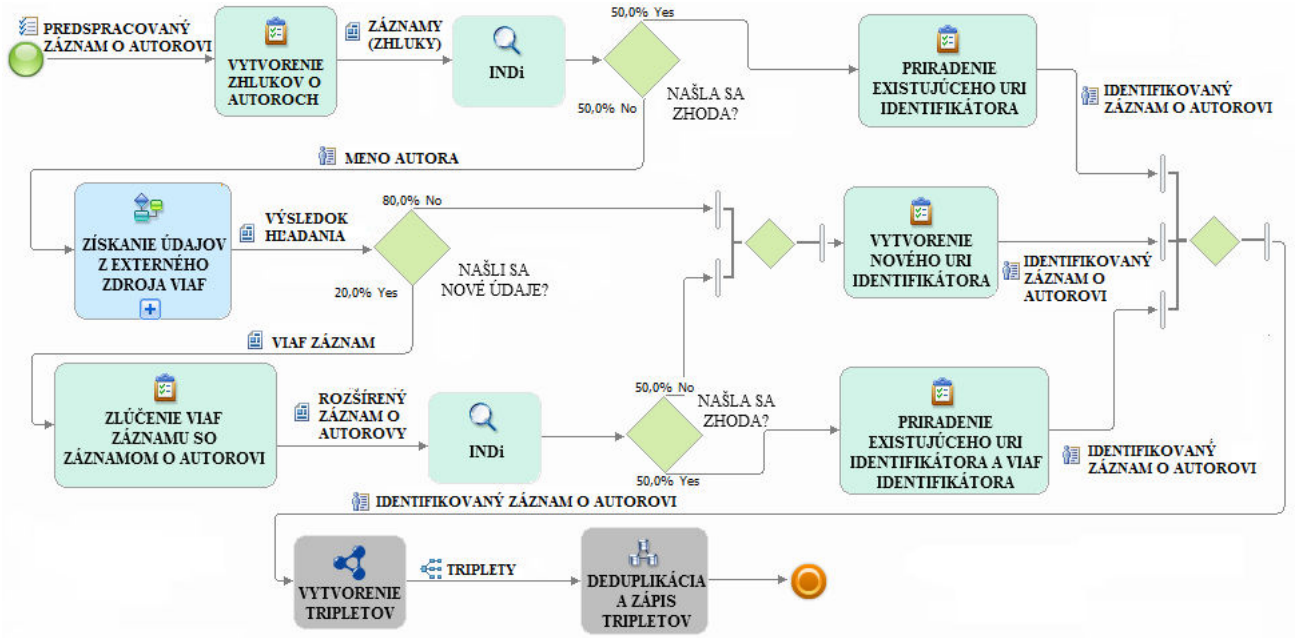
V prípade, keď autor A nedosiahne minimálnu hranicu podobnosti so žiadnym autorom z VIVO modelu, alebo ak dosiahne rovnakú podobnosť s viacerými autormi z VIVO modelu, tak je potrebné zapojiť do procesu informácie z externého systému VIAF.

Informácie z externého zdroja použijeme na obohatenie existujúceho záznamu o autorovi A, čím sa zvyšuje pravdepodobnosť nájdenia zhody s niektorým autorom z VIVO modelu. Následne sa uskutoční rovnaké porovnanie ako v kroku 4.

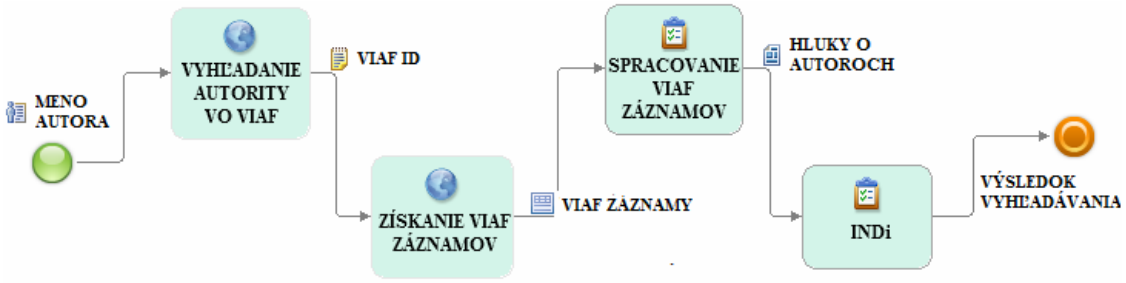
Ak sa ani po tomto vyhodnotení nenájde zhoda medzi autorom A a niektorým z vybraných autor z kroku 2, tak sa daný autor ešte nenachádza vo VIVO modeli a systém mu vygeneruje nový URI identifikátor.

6. **Tento proces sa opakuje pre všetkých autorov z bibliografického záznamu**

Celý proces identifikácie autora s použitím externého zdroja informácií je znázornený na obrázku č.7.



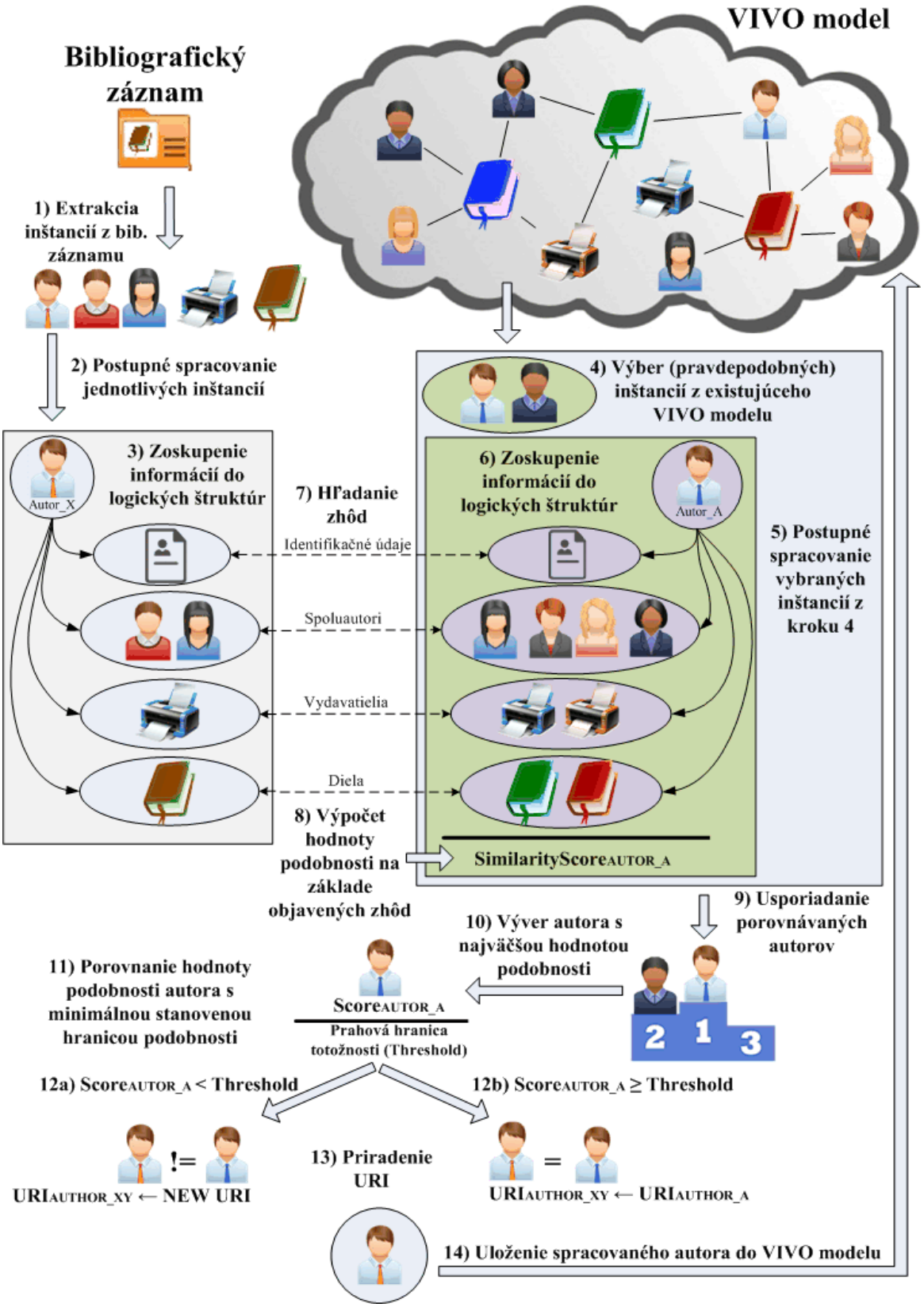
Obrázok č. 7. Biznis proces znázorňujúci postup identifikacie autorov.



Obrázok č. 8. Podproces Získavanie obohacujúcich informácií z externého zdroja VIAF.

Vyhľadávanie autority v databáze VIAF nie je vždy jednoznačné. Vyhľadáva sa na základe mena autora, čo predstavuje rovnaký princíp ako vyhľadávanie autora vo VIVO modeli. Preto na získanie najpravdepodobnejšieho autora vo VIAF databáze použijeme INDi algoritmus. INDi algoritmus využije nastavené parametre porovnávania a koeficienty podobnosti. Zmeníme sa iba hranice podobnosti, pretože kvalita a obsah údajov vo VIAFe môže byť iná ako kvalita práve spracovávaných bib. záznamov. Daná hodnota bude konfigurovateľná v systéme.

Proces identifikácie autora (bez použitia informácií z externého zdroja VIAF) je podrobnejšie znázornený na obrázku č. 9. Na vstupe sa nachádza bibliografický záznam, ktorý obsahuje inštancie autorov a VIVO model, do ktorého sa majú dané inštancie zaradiť. Identifikácia inštancie jej a zaradenie do modelu obsahuje 14 krokov. Tento proces sa opakuje pre všetkých autorov z bibliografického záznamu.



Obrázok č. 9 Proces identifikácie autorov.

Keď proces rozšírime na inštancie všetkých definovaných tried (aj diela a vydavateľov), dostaneme univerzálny identifikačný proces. Identifikačné procesy sa budú medzi sebou líšiť zoznamom vybraných parametrov pri hľadaní zhôd, minimálnou hranicou podobnosti a čiastočne v zoskupení informácií do logických štruktúr, použitých v kroku 3 a 6.

3.4.2 Proces identifikácie diela

Proces identifikácie diela je analogický s predchádzajúcimi procesmi.

1. **vytvorenie záznamu o diele**

Vytvoríme štruktúru (vlastný záznam), ktor bude obsahovať informácie o vybranom diele (nazvime ho D) z práve spracovaného bib. záznamu. Štruktúra obsahuje názov diela, jeho paralelné názvy a podnázvy, miesto a čas vydania, publikačné identifikátory ISBN alebo ISSN v závislosti od typu diela. Záznam obsahuje taktiež zoznam autorov a vydavateľov podieľajúcich sa na vytvorení diela.

2. **vytvorenie porovnávacích VIVO záznamov (zhlukov) pre diela**

VIVO záznamy čerpané z VIVO modelu sa vyhľadávajú na základe názvu diela, jeho paralelných názvov, ISBN a ISSN identifikátorov. Každý nájdený záznam predstavuje jedno dielo, ktoré obsahuje identifikačné údaje a zoznam svojich autorov a vydavateľov.

3. **proces porovnania**

Poradie porovnávaných prvkov je nasledovné:

Priorita	Názov
1.	Zhoda v hlavnom názve diela*
2.	Zhoda v hlavnom názve diela (NT)
3.	Zhoda v paralelnom názve diela*
4.	Zhoda v paralelnom názve diela (NT)
5.	Zhoda v podnázve**
6.	Zhoda v ISBN/ISSN
7.	Zhoda v mieste vydania
8.	Zhoda v roku vydania
9.	Zhoda vo vydavateľovi***
10.	Zhoda v autoroch, editoroch, prekladateľoch****

Poznámka:
* Ak sa nenašla zhoda, vykonáme porovnanie s použitím JWD.

** Porovnávací reťazec má tvar: "hlavný názov diela : podnázov"

***Vydavateľov sa porovnávame na základe: identifikátorov (ak sú priradené), názvu a variantných foriem názvu s použitím s použitím JWD a ak nebola objavená zhoda, tak s použitím NT.

**** Osoby porovnávame na základe URI identifikátora, keďže rozpoznávanie diela je vykonávané v čase, keď sú už všetky osoby z bib. záznamu identifikované a je im priradený URI identifikátor v rámci VIVO modelu.

Na základe objavených zhôd vypočítame hodnotu (skóre) podobnosti medzi dielom D a vybranými dielami z VIVO modelu. Použijeme vzorec 1.4.a.

4. vyhodnotenie a priradenie URI identifikátora

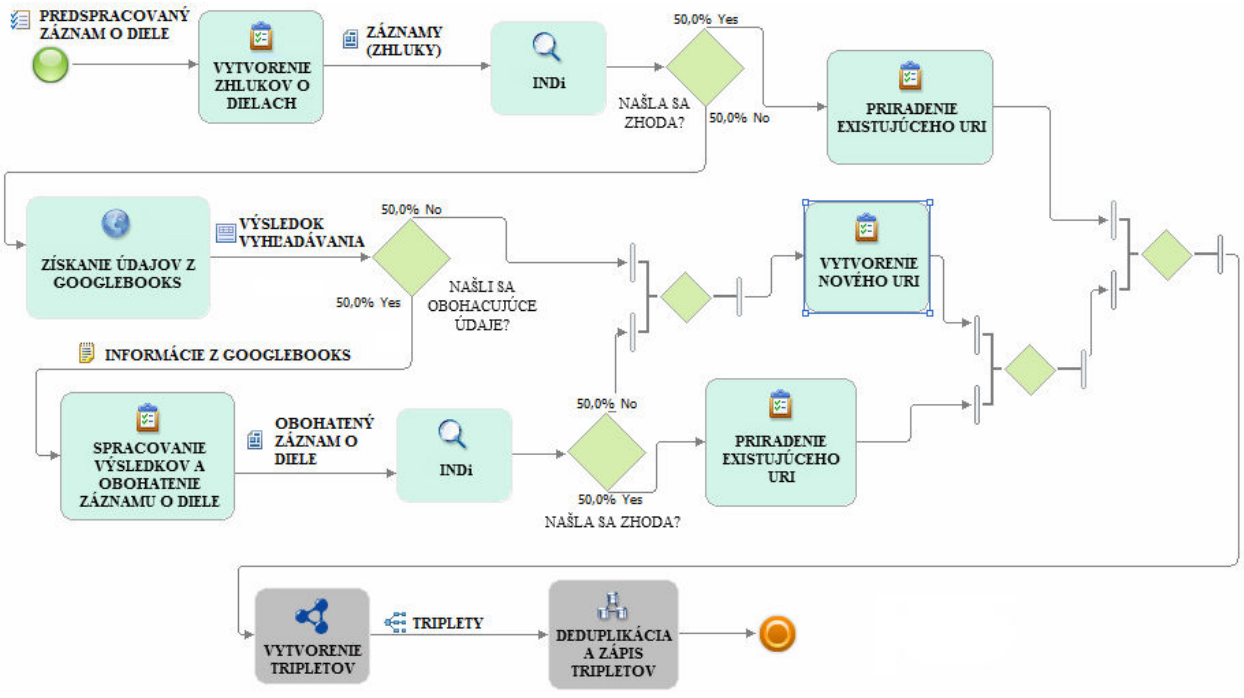
Ak dielo z najvyššou hodnotou podobnosti, z kroku 3, dosiahlo minimálnu stanovenú bodovú hranicu podobnosti s dielom D, tak môžeme dielu D priradiť URI identifikátor objaveného diela z VIVO modelu. Dielo D sa už nachádza v modeli. V opačnom prípade dielu D priradíme nový URI identifikátor.

5. získanie informácií z externého systému GoogleBooks

V prípade, keď sa nám nepodarí nájsť zhodu s dielom vo VIVO modeli, zapojíme do procesu identifikácie informácie z externého systému GoogleBooks.

Informácie z GoogleBooks použijeme na doplnenie informácií o diele z bib. záznamu, čím sa zvyšuje pravdepodobnosť nájdenia zhody s niektorým z diel vo VIVO modeli.

Následne uskutočníme rovnaké porovnanie ako v kroku 4. Ak ani po tomto vyhodnotení nenájdeme podobné dielo vo VIVO modeli, tak dielu z kroku 1 priradíme nový URI identifikátor.



Obrázok č. 10. Biznis proces identifikácie inštancie diela

3.4.3 Proces identifikácie vydavateľov

Proces identifikácie vydavateľov je analogický s procesmi identifikácie autorov a diel. Centrálnou entitou je v tomto prípade vydavateľ.

Postup:

1. **vytvorenie záznamu o vydavateľovi z bib. záznamu**

Častí záznamu:

vydavateľ — meno vydavateľa a identifikátor v rámci knižničného systému, z ktorého bol záznam získaný. V zázname sa nachádzajú aj variantné formy názov vydavateľa a miesto/sídlo.

autori — autori z bib. záznamu, ktorí nesú zodpovednosť za vytvorenie diel, ktoré boli vydané daným vydavateľom.

diela — diela, ktoré vydal daný vydavateľ.

2. **vytvorenie porovnávacích záznamov z VIVO modelu**

Na základe názvu vydavateľa z bib. záznamu sa vyhľadáme všetkých vydavateľov s rovnakým alebo variantným názvom v existujúcom VIVO modeli. Každý vybraný vydavateľ z VIVO modelu so sebou nesie aj zoznamy autorov, s ktorými spolupracoval a zoznam diel, ktoré publikoval.

3. **proces porovnania (INDI algoritmus)**

INDI algoritmus v tomto prípade použijeme na porovnávanie vydavateľa z bib. záznamu a vydavateľov z VIVO modelu získaných v kroku 2.
Poradie porovnávaných prvkov je nasledovné:

Priorita	Názov
1.	Zhoda v názve vydavateľa*
2.	Zhoda v názve vydavateľa (NT)
3.	Zhoda v alternatívnom názve vydavateľa*
4.	Zhoda v alternatívnom názve vydavateľa (NT)
5.	Zhoda v identifikátore z iného systému
6.	Zhoda v dielach**
7.	Zhoda v spolupracujúcich autoroch**

Poznámka:

* ak nebola objavená zhoda, nastáva porovnávanie s použitím JWD.

** Diela porovnávame na základe URI identifikátorov, keďže už boli identifikované v predchádzajúcom procese a majú pridelený svoj URI identifikátor.

*** Spoluautorov porovnávame na základe priradených URI identifikátorov. Rovnako ako diela aj osoby už boli identifikované v predchádzajúcom procese.

Každému porovnávanému záznamu priradíme bodové ohodnotenie podľa vzorca 1.4.a. Týmto spôsobom sa vyhodnotia všetky VIVO záznamy.

4. **vyhodnotenie a priradenie URI identifikátora**

Ak vydavateľ z bib. záznamu, nazvime ho V, dosiahol podobnosť s niektorým vydavateľom z VIVO modelu nazvime ho B, ktorá presahuje stanovenú bodovú hranicu podobnosti a neexistuje iný vydavateľ vo VIVO modeli s rovnakou hodnotou podobnosti ako B, tak danému vydavateľovi V priradíme URI identifikátor vydavateľa B z VIVO modelu. Inak je vydavateľovi V pridelený nový URI identifikátor.
Pri identifikácií vydavateľov v súčasnosti nepoužívame žiadne externé zdroje informácií.

3.5 Externé zdroje

V kapitole je uvedený spôsob získavania informácií z externých systémov VIAF a GoogleBook spolu so zoznamom informácií, ktoré použijeme v procesoch identifikácie inštancií osôb a diel.

3.5.1 VIAF

Dopyt je vzniká vyskladaním http GET požiadavky, ktorá je zaslaná na server VIAF.

http://www.viaf.org/viaf/AutoSuggest?query=<*priezvisko*, *meno*>

alebo http://www.viaf.org/viaf/AutoSuggest?query=<*priezvisko* >

Napr.: http://www.viaf.org/viaf/AutoSuggest?query=Loderer

Výsledkom je JSON súbor obsahujúci výsledky vyhľadávania.

```
{"query":"loderer","result":[{result_set}]
```

Pole result obsahuje záznamy prislúchajúce k hľadanému pojmu. V záznamoch sa nachádzajú združené identifikátory z rôznych inštitúcií.

Napr.:

```
{"query":"loderer","result":
        [{"term":"Loderer, Benedikt, 1945-",
          "lc":"nr99011936",
          "dnb":"124169635",
          "bnf":"15021227",
          "viafid":"49147749"}]
}
```

Zo získaného záznamu vyberieme identifikátor `viafid` hľadanej osoby. V niektorých prípadoch sa môže vo VIAF databáze nachádzať viac osôb s rovnakým menom. V tom prípade obsahuje záznam zoznam osôb a k ním prislúchajúci `viafid` identifikátor. Následne pre každú osobu obsiahnutú v zázname formulujeme ďalší http GET dotaz, ktorým získame úplný viaf záznam o danej osobe.

http://viaf.org/viaf/?<*viaf_Identifikátor*>/marc21.xml

Napr.: http://viaf.org/viaf/?49147749/ marc21.xml

Zo získanej odpovede extrahujeme polia uvedené v tabuľke č. 2.

Tabuľka č. 2 Zoznam polí a informácií extrahovaných z VIAF záznamu.

Pole	Údaj	Opakovateľnosť
024 a	Viaf URI	nie
700 a	Meno autora (rôzne formy)	áno
950 a	Mená spoluautorov	áno
910 a	Názvy diel	áno
901 a	ISBN	áno
921 a	Vydavatelia spolupracujúci s daným autorom	áno

3.5.2 GoogleBooks

GoogleBooks poskytuje webové API pre vyhľadávanie a prácu so záznamami o dielach. Vyhľadávanie sa uskutočňuje zaslaním http GET požiadavky na server. Výsledkom vyhľadávania je súbor vo formáte JSON, v ktorom sú zapísané informácie o počte nájdených kníh a informácie jednotlivých knihách. Zo získaných záznamov sa využijú informácie uvedené v tabuľke č. 3.

Tabuľka č. 3 Použité hodnoty z externého systému GoogleBooks.

Názov atribútu	Hodnota v atribúte
selfLink	identifikátor knihy v systéme GoogleBooks
volumeInfo.subtitle	podnadpis diela
volumeInfo.publishedDate	dátum publikovania
volumeInfo.pageCount	počet strán
volumeInfo.language	jazyk publikácie
volumeInfo.description	abstrakt, popis
volumeInfo.industryIdentifiers.identifier	identifikátory ISBN10 a ISBN13

V systéme vytvárame dva typy vyhľadávacích dotazov:

1) Vyhľadávanie pomocou ISBN

https://www.googleapis.com/books/v1/volumes?q=isbn:<ISBN>, kde sa reťazec <ISBN> nahradí hodnotou ISBN hľadanej knihy.

Napr.: https://www.googleapis.com/books/v1/volumes?q=isbn:808903330X

2) Vyhľadávanie pomocou názvu diela a mien autorov

https://www.googleapis.com/books/v1/volumes?q=<TITLE>+inauthor:<AUTHOR$_1$>+inauthor:<AUTHOR$_2$>+....+inauthor:<AUTHOR$_n$>, kde reťazec <TITLE> nahradíme hlavným názvom diela a reťazce <AUTHOR$_1$>...<AUTHOR$_n$> menami autorov diela v tvare meno+priezvisko, prípadne iniciála mena+priezvisko.

Napr.: https://www.googleapis.com/books/v1/volumes?q=Piata+hora+inauthor:Coelho+Pablo

Získané informácie použijeme na obohatenie záznamu o diele v procese identifikácie diela. Získané informácie sa uložia do VIVO modelu.

3.6 Spresnenie vyhľadávania

Vyhľadávanie a porovnávanie textových reťazcov doplníme vyhľadávaním s použitím miery podobnosti. V bib. záznamoch sa môžu nachádzať mená autorov, diel a iné textové informácie, kde mohlo dôjsť ku chybe pri ich zápise alebo editácií.

Vyhľadávanie a porovnávanie textových reťazcov budeme realizované pomocou metriky Jaro-Winkler distance. Túto metriku aplikujeme aj na vyhľadávanie inštancií v rámci VIVO modelu. Keďže dopytovací jazyk SPARQL neobsahuje implementáciu danej metódy, musíme ju vytvoriť.
Metódu je možné zapísať nasledujúcim pseudokódom:

Vstup: porovnávané reťazce s1, s2 a používateľom definovaná minimálna hranica podobnosti t.

Výstup: hodnota true (reťazce sú podobné) alebo false (reťazce nie sú podobné).

```
1: a← JaroWinklerDistance(s1,s2);
2: if  a >= t then return true
3: else       return false
```

3.7 Vyriešenie problému SAME AS vo VIVO modeli

V časti analýza bolo uvedené, že použitá verzia VIVO modelu 1.5 neobsahuje vzťah SAME AS. Z toho dôvodu nemôžeme každej inštancii prideliť vlastný URI identifikátor, zaradiť ju do existujúceho modelu a až potom zisťovať, či práve zaradená inštancia nie je duplicita, ktorú môžeme spojiť s inou existujúcou inštanciou z modelu, pomocou vzťahu SAME AS. Inštanciu musíme identifikovať ešte pred zaradením do modelu.

Toto obmedzenie sme vyriešili úpravou VIVO modelu, resp. upravili sme logický návrh zlučovania a zápisu údajov pre jednotlivé inštancie. Vo VIVO modeli sme použili vlastnosť `dcterms:alternative`.

Štandardný spôsob zápisu pomocou vzťahu SAME AS by vyzeral nasledovne

Existujúci VIVO model:

```
<rdf:Description rdf:about="http://ml.fiit.stuba.sk/individual/person1">
    <foaf:lastName>Krakovský</foaf:lastName>
    <foaf:firstName>Alexander</foaf:firstName>
    <myont:yearOfBirth>1942</myont:yearOfBirth>
</rdf:Description>
```

Systém potrebuje pridať do existujúceho VIVO modelu nový spracovaný bib. záznam:

```
<rdf:Description rdf:about="http://ml.fiit.stuba.sk/individual/person2">
    <foaf:lastName>Krakovsky</foaf:lastName>
    <foaf:firstName>A.</foaf:firstName>
    <myont:yearOfBirth>1942</myont:yearOfBirth>
</rdf:Description>
```

Pridaná inštancia person2 je totožná s existujúcou inštanciou person1. Obom inštanciám sa pridá potrebný vzťah. Vzhľad výsledného modelu bude vyzerať:

```
<rdf:Description rdf:about="http://ml.fiit.stuba.sk/individual/person1">
    <foaf:lastName>Krakovský</foaf:lastName>
    <foaf:firstName>Alexander</foaf:firstName>
    <myont:yearOfBirth>1942</myont:yearOfBirth>
    <owl:sameAs rdf:resource="http://ml.fiit.stuba.sk/individual/person2"/>
</rdf:Description>
<rdf:Description rdf:about="http://ml.fiit.stuba.sk/individual/person2">
    <foaf:lastName>Krakovsky</foaf:lastName>
    <foaf:firstName>A.</foaf:firstName>
    <myont:yearOfBirth>1942</myont:yearOfBirth>
    <owl:sameAs rdf:resource="http://ml.fiit.stuba.sk/individual/person1"/>
</rdf:Description>
```

Naše riešenie s použitím vzťahu `dcterms:alternative`

VIVO model:

```
<rdf:Description rdf:about="http://ml.fiit.stuba.sk/individual/person1">
    <foaf:lastName>Krakovský</foaf:lastName>
    <foaf:firstName>Alexander</foaf:firstName>
    <myont:yearOfBirth>1942</myont:yearOfBirth>
</rdf:Description>
```

Systém potrebuje pridať inštanciu autora (A. Krakovsky, 1942), ktorú vyhodnotil ako totožnú s inštanciou vo VIVO modeli. Novo priraďovaná inštancia sa líši v mene autora, preto sa použije vlastnosť dcterms:alternative. Výsledný model:

```
<rdf:Description rdf:about="http://ml.fiit.stuba.sk/individual/person1">
    <foaf:lastName>Krakovský</foaf:lastName>
    <foaf:firstName>Alexander</foaf:firstName>
    <myont:yearOfBirth>1942</myont:yearOfBirth>
    <dcterms:alternative>Krakovsky, A.</dcterms:alternative>
</rdf:Description>
```

Prvý spôsob zápisu je robustnejší, redundantný a umožňuje jednoduchšie zlučovanie a rozdeľovanie nesprávne priradených inštancií. Na druhej strane vyžaduje určiť primárnu inštanciu, s ktorou sa budú porovnávať spracovávané bibliografické záznamy, čo môže viesť v určitých prípadoch k logickým chybám.

Druhý spôsob je úspornejší, nedochádza k takmer žiadnej redundancií údajov a rovnaké inštancie nemajú pridelený samostatný URI identifikátor. Problémom je zlučovanie inštancií, kedy sa musia všetky údaje druhej inštancie ručne zlúčiť s údajmi prvej inštancie a nakoniec odstrániť druhú inštanciu. Pri rozdelení jednej inštancie na dve, je potrebné ručne vytvoriť druhú inštanciu a ďalšia ručná separácia údajov z prvej inštancie do druhej.

Použitý druhý spôsob nie je výhodnejší, najmä z pohľadu ručného zlučovania a rozdeľovania inštancií, je však navrhnutý ako dočasné riešenie, kým VIVO systém nebude obsahovať implementáciu vzťahu SAME AS.

3.8 Úprava VIVO modelu

Niektoré údaje z bib. záznamov nemali v použitom VIVO modeli ekvivalentný atribút, ktorý by vyjadroval ich sémantický význam. Preto sme VIVO model rozšírili o dané atribúty. Vytvorené atribúty sú súčasťou našej vlastnej ontológie s označením myont : http://vivo.stuba.fiit.dp/.

Tabuľka č. 4 Zoznam pridaných atribútov vo VIVO modeli

Názov atribútu	Účel	URI	Obmedzenie na triedu a podtriedy	Typ
subtitle	podnadpis diela	http://vivo.stuba.fiit.dp/subtitle	bibo:Document	String
yearOfBirth	rok narodenia osoby	http://vivo.stuba.fiit.dp/yearOfBirth	foaf:Person	Year(YYYY)
deathYear	rok úmrtia osoby	http://vivo.stuba.fiit.dp/deathYear	foaf:Person	Year(YYYY)
viaf	VIAF identifikátor	http://vivo.stuba.fiit.dp/viaf	foaf:Person	URI/URL
udc	medzinárodného desatinného triedenia	http://vivo.stuba.fiit.dp/udc	bibo:Document	String

V dvoch prípadoch využijeme na zápis údajov atribúty, ktoré VIVO model obsahuje, ale štandardne ich nevyužíva. Zoznam atribútov je uvedený v tabuľke číslo 5.

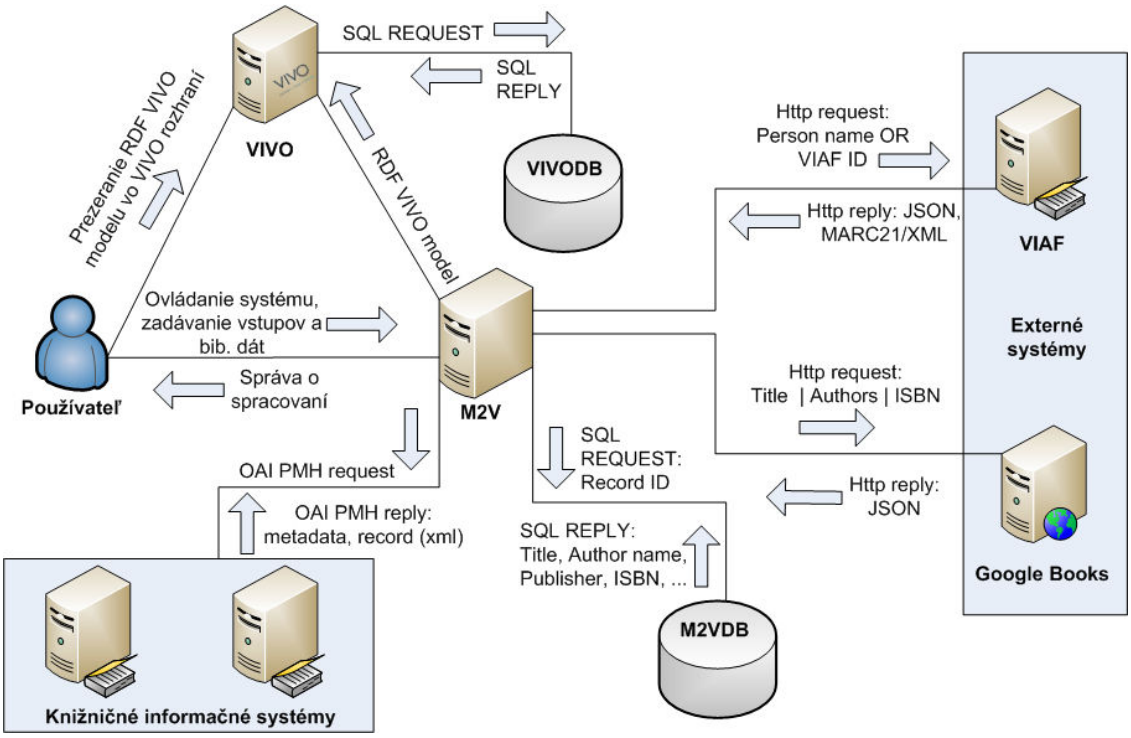
Tabuľka č. 5 Zoznam špeciálne použitých atribútov.

Názov atribútu	Účel	URI	Obmedzenie na triedu a podtriedy	Typ
alternative	Alternatívne meno alebo názov	http://purl.org/dc/terms/alternative	owl:Thing	String
source	Elektronická lokácia a prístup k dokumentu	http://purl.org/dc/terms/source	bibo:Document	URI/URL

3.9 Architektonický návrh

Systém sme navrhli ako program, ktorý bude pracovať na jednej pracovnej stanici. Systém bude využívať internú databázu s názvom M2VDB, v ktorej budú uložené konfiguračné nastavenia, hodnoty potrebné pre správne generovanie URI adries a všetky bibliografické záznamy, ktoré sú zaradené do procesu predspracovania, pred samotnou transformáciou do VIVO modelu.

Výstup systému, ktorým je VIVO model obsahujúci spracované bibliografické záznamy, je možné prezerať pomocou grafického rozhrania systému VIVO. Systém VIVO môže byť na rovnakej alebo inej pracovnej stanici ako systém M2V. Na obrázku č. 11 je systém VIVO zobrazený ako samostatná pracovná stanica, na ktorej si používateľ môže prezerať výsledok transformácie(sémantizácie) bib. záznamov.



Obrázok č. 11. Architektúra systému s typmi komunikácie medzi komponentmi.

3.9.1 Komunikácia systémov

Systém bude využívať viacero typov komunikácie. Vstupné príkazy bude zadávať používateľ prostredníctvom grafického používateľského rozhrania. Systém bude komunikovať s internou databázou M2VDB, v ktorej budú uložené konfiguračné nastavenia a všetky bibliografické záznamy pripravené na spracovanie. Databáza bude slúžiť ako vstupný zásobník pre proces spracovania/sémantizácie.

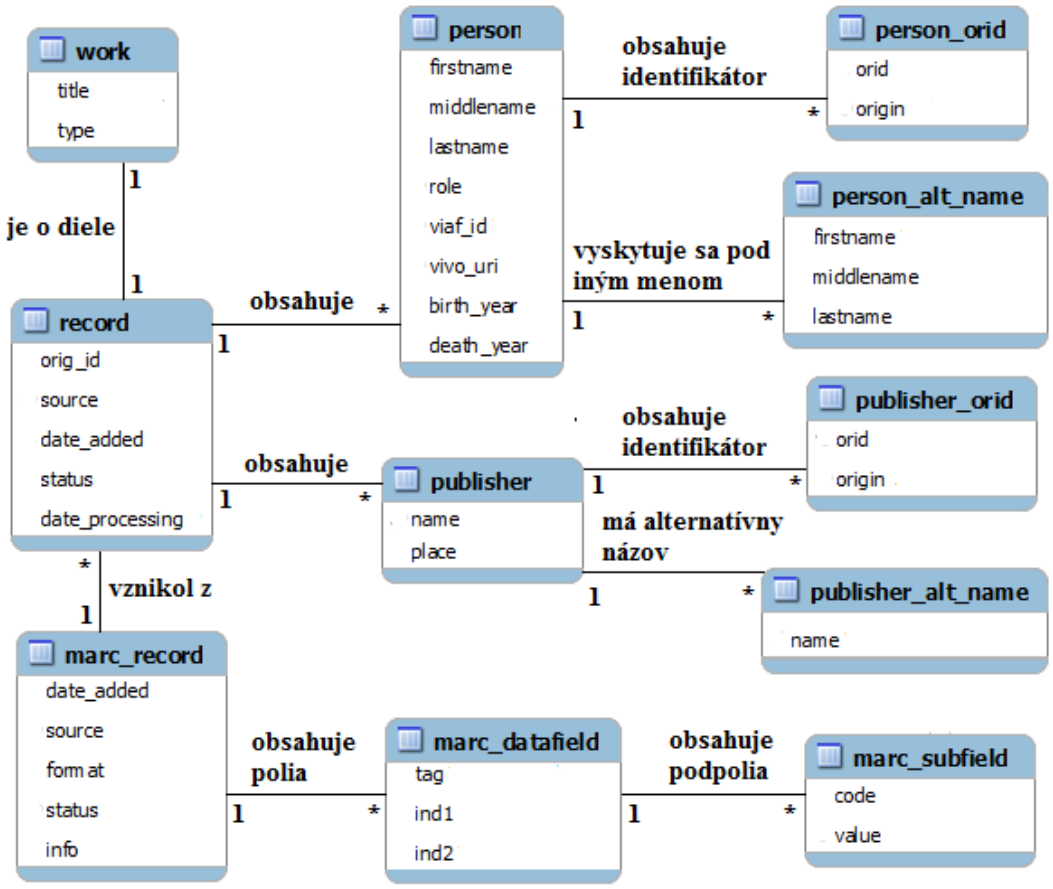
Systém bude môcť získavať bibliografické záznamy importovaním súborov, ktoré zadá používateľ alebo priamo z digitálnych knižníc pomocou protokolu OAI PMH alebo Z39.50. Získané záznamy sa predspracujú a uložia do databázy M2VDB, odkiaľ budú čerpané v procese sémantizácie.

Systém bude získavať pomocné identifikačné a obohacujúce dáta z externých systémov VIAF a GoogleBooks. Obidva externé zdroje sa dopytujú zaslaním http požiadavky (http request). Odpoveďou je súbor vo formáte json alebo xml, obsahujúci požadované informácie alebo informáciu o nenájdení požadovaných údajov.

Výstupom systému M2V bude VIVO model vo formáte RDF/XML, obsahujúci spracované bibliografické záznamy. Výstupný model bude slúžiť ako vstup pre VIVO systém. Systém VIVO potrebuje pre svoje fungovanie vlastnú databázu (VIVO_DB). Model sa do systému VIVO nebude neprenášať automaticky. Import bude musieť vykonať používateľ, po skončení transformácie bib. záznamov.

3.10 Logický model

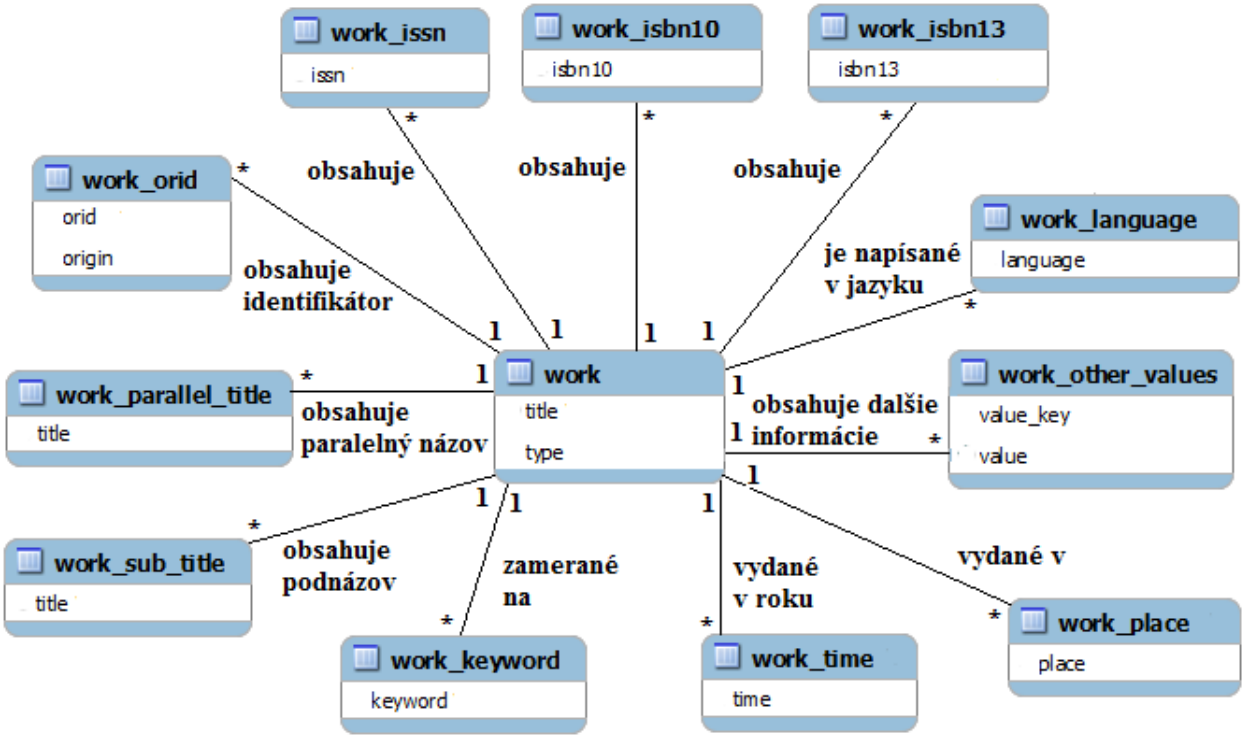
Navrhovaný systém bude ukladať všetky nahrané bibliografické záznamy do databázy, ktorá bude slúžiť ako zásobník v procese predspracovania. Používateľ bude môcť v tejto fáze dáta prezerať, upravovať alebo úplne odstrániť. Všetky tieto informácie budú uložené v databáze s názvom M2VDB. Databáza môže byť charakterizovaná ako dátový sklad, v ktorom sú uložené bibliografické dáta získané z knižničných systémov. Centrálnou tabuľkou bude tabuľka marc_record, ktorá združuje informácie získané z načítaných bib. záznamov. Ďalšími úpravami sa z bibliografických dát v procese predspracovania extrahujú informácie o jednotlivých entitách dielo, osoba a vydavateľ. Centrálnou tabuľkou združujúcou dané informácie je tabuľka record.



Obrázok č. 12 Logický model databázy M2VDB

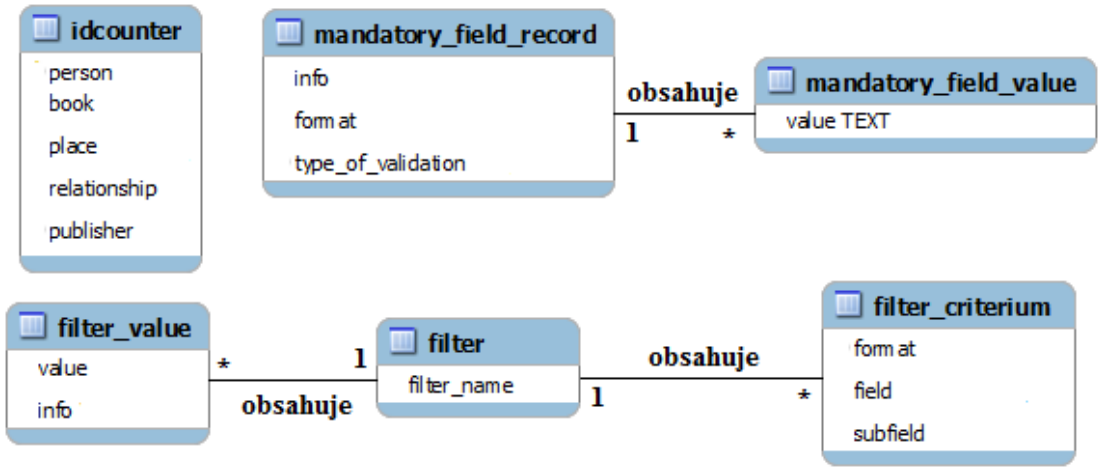
Logický model obsahuje entity a ich kardinality. Vzťahy medzi entitami sú opísané slovne. Slovné popisy je potrebné čítať v smere hodinových ručičiek, napr. record „obsahuje“ person , alebo record „je o diele“ work.

Entita work (dielo) združuje veľké množstvo údajov získaných zo spracovania bibliografických údajov. Kvôli prehľadnosti je daná entita znázornená samostatným diagramom na obrázku č. 13.



Obrázok č. 13 Diagram znázorňujúci entitu work a jej vzťahy s inými entitami.

Súčasťou systému sú aj entity, ktoré uchovávanú konfiguračné nastavenia a pravidlá na kontrolu kvality obsahu bibliografických záznamov.



Obrázok č. 14 Logický model znázorňujúci entity uchovávajúce konfiguračné nastavenia systému.

3.10.1 Opis dátového modelu

V kapitole sa nachádza zoznam entít a ich krátky popis.

marc_record entita, ktorá združuje informácie o nespracovanom bib. zázname. Jeden záznam v tabuľke marc_record predstavuje jeden bibliografický MARC záznam.

marc_datafield obsahuje údaje uložené v štruktúre pole (*datafield*) v MARC záznamoch. Entita slúži aj na zápis informácií zo štruktúry riadiace pole (*controlfield*). V tom prípade sa neuvádza hodnota pre ind1 a ind2.

marc_subfield obsahuje informácie zo štruktúry podpole (*subfield*). Do tabuľky sa zapíše iba názov podpoľa a jeho hodnota.

Transformáciou údajov z predchádzajúcich entít, vznikajú údaje, ktoré zapisujeme do nasledujúcich entít:

record entita, ktorá združuje informácie o predspracovanom bib. zázname. Entita record vzniká predspracovaním entity marc_record.

work entita obsahuje informácie o diele. Atribút type určuje formu diela: akademický článok, článok, kniha, skriptum, atď.

person obsahuje základné informácie o osobe (autorovi, editorovi, prekladateľovi diela), meno, identifikátory, rok narodenia a úmrtia, ktoré sa extrahovali z bib. záznamu.

publisher informácie o vydavateľovi diela (jeho názov a miesto).

Entity, ktorých názvy končia na príponu **_orid,** obsahujú zoznam identifikátorov, pod ktorými inštancia danej entity vystupuje v externých systémoch odkiaľ je inštancia získaná alebo kde bola identifikovaná. Entity, ktorých názvy končia na príponu **_alt_name** majú okrem hlavného názvu aj variantnú formu názvu (napr.: iná forma mena osoby, alebo rôzne mená vydavateľov: Slovenská poľnohospodárska univerzita a Slovenská poľnohospodárska univerzita v Nitra).

Entita work spája informácie o vybranom diele. Z uvedených názvov je zrejmé aké informácie sa nachádzajú v jednotlivých entitách. Nejasný môže byť význam entity **work_other_values**. Entita uchováva informácie, ktoré sa nedali zaradiť do niektorej z existujúcich entít (napr.: počet strán, odkaz na elektronické médium, číslo vydania, a iné). Informácie sú vždy reprezentované ako dvojica: názov hodnoty a samotná hodnota.

Opis a význam entít, ktoré slúžia na uchovávanie používateľských nastavení a pravidiel na kontrolu kvality:

idcounter entita potrebná na generovanie unikátnych URI identifikátorov vytváraných pri spracovaní bib. záznamov. Entita uchováva hodnotu najvyššieho použitého poradového čísla pre každý typ generovaný typ URI identifikátora.

filter entita slúži na zadefinovanie validačného pravidla, pri kontrole kvality zameranej na validáciu hodnôt podľa stanoveného číselníka.

filter_criterium slúži na uchovávanie zoznamu polí a podpolí, ktoré sa majú kontrolovať v procese kontroly kvality.

filter_value slúži ako číselník hodnôt pre validáciu vybraných polí a podpolí .

mandatory_field_record obsahuje informácie na vytvorenie validačného pravidla, pri kontrole kvality, zameranej na prítomnosť definovaných polí a podpolí. Tabuľka bude slúžiť ako číselník pre konfiguráciu daných pravidiel. Tabuľka bude uchovávať informácie systémového charakteru. Bežný používateľ nebude mať možnosť meniť tieto nastavenia.

mandatory_field_value obsahuje zoznam polí a podpolí pre jednotlivé MARC formáty, ktorých prítomnosť v bib. zázname sa má skontrolovať. Tieto údaje definuje používateľ.

Fyzický model databázy M2VDB sa nachádza v časti implementácia, kapitola 4.3.1 Fyzický model.

4 Implementácia

V tejto kapitole je opísaný spôsob riešenia a postupy použité pri realizácií systému. Nachádza sa tu bližší opis vybraných častí systému spolu s technológiami, ktoré boli použité pri jeho tvorbe. Celý systém je vyvíjaný v programovacom jazyku Java SE (JDK 1.7.0_51). Systém bol implementovaný podľa návrhu z predchádzajúcej kapitoly. Niektoré časti systému neboli z časových dôvodov implementované.

4.1 Grafické používateľské rozhranie

Používateľské rozhranie je realizované pomocou technológie Java FX. Java FX je štandardnou súčasťou balíka Java JDK aj JRE od verzie 8. Do verzie Java 7 je potrebná knižnica *jfxrt.jar*, ktorá sa vydáva pod označením JavaFX SDK. Ukážka grafického používateľského rozhrania sa nachádza v kapitole B Používateľské rozhranie. Pri tvorbe rozhrania boli použité komponenty z balíka TiwulFX.

Jazyková lokalizácia je realizovaná technológiou Resource Bundles a jazykových lokalizačných súborov .properties. Konfiguračný súbor pre slovenský jazyk Language_sk.properties sa nachádza v priečinku <*projekt*>/src/resources/.

4.2 Spracovanie bibliografických záznamov

Bibliografické záznamy do systému importujeme vo formáte marc/xml. Spracovanie je realizované technológiou DOM. Pôvodne navrhovanú technológiu SAX, ktorá by umožňovala prúdové spracovanie veľkého objemu dát, sme kvôli špeciálnym znakom v bibliografických záznamoch nemohli použiť. Z toho dôvodu je potrebné veľké marc/xml súbory (cca nad 400 MB) rozdeliť na menšie súbory.

V triede sk.stuba.fiit.ml.dp.preprocessing.MarcXMLParser.java sme implementovali metódy k spracovaniu vybraných marc/xml záznamov.

Bibliografické dáta sú spracovávané a ukladané v databáze do tabuliek marc_record, marc_datafield a marc_subfield. Informácie z údajovej štruktúry Controlfield sa pri prevode do databázy ukladajú do tabuľky marc_datafield. V tomto prípade sa neuvádza hodnota tagu ani hodnoty indikátorov.

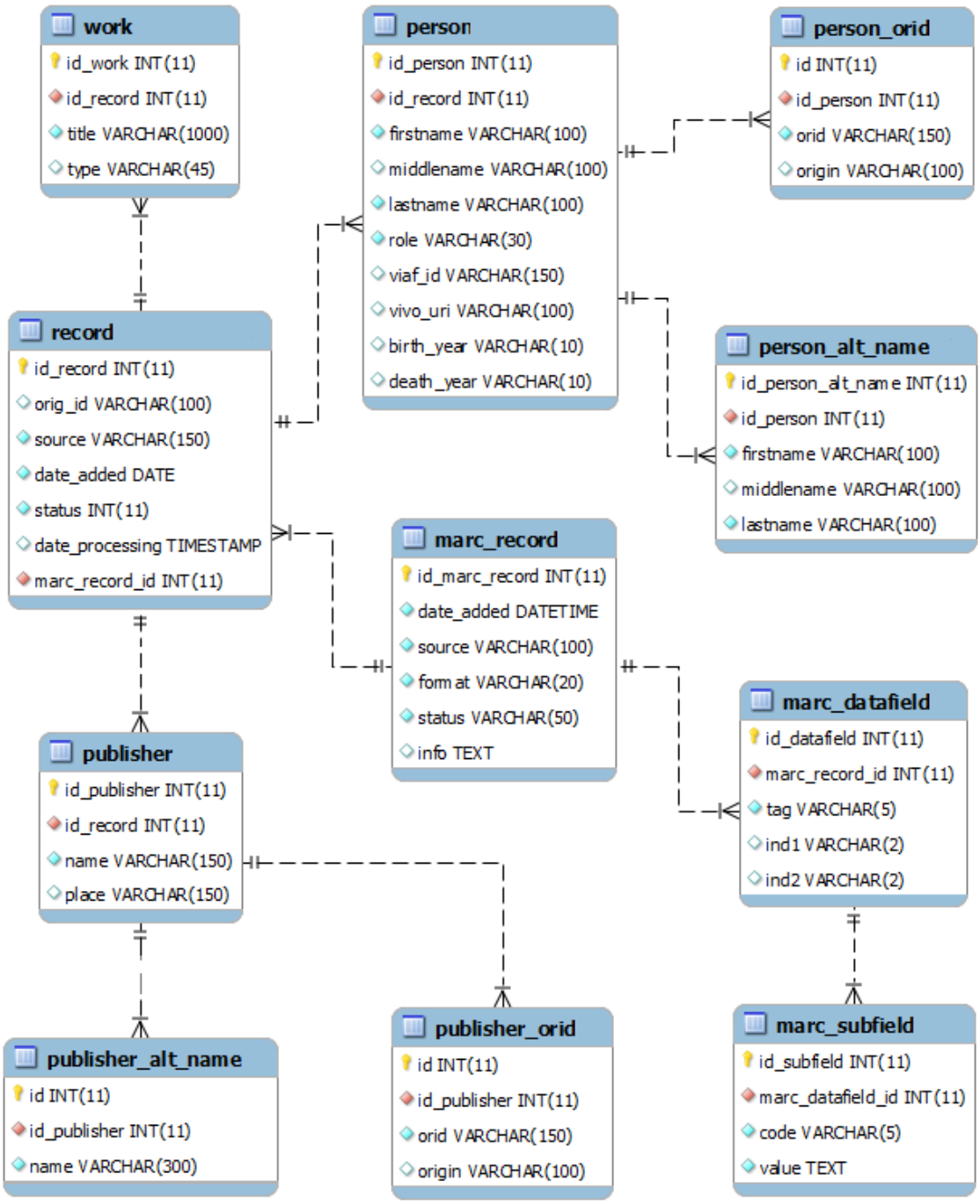
4.3 Databáza

Pri implementácii sme použili relačnú databázu MySQL, na ktorú sme systém M2V pripojili pomocou jdbc konektora. K spracovaniu údajov v databáze sme použili softvérový rámec Hibernate, ktorý nám umožnil objektovo-relačné mapovanie a jednoduchšiu manipuláciu s dátami. V programe využívame mapovanie entít pomocou anotácií (@). Konfiguračný súbor hibernate.cfg.xml sa nachádza v priečinku <*projekt*>/src/resources/.

Prístupové nastavenia sú uložené v súbore dbconfig.properties v priečinku <*projekt*>/resources/. Pri spustení programu sú databázové nastavenia z tohto súboru prenesené do nastavení hibernate.cfg.xml. Triedy, ktoré sme pomocou rámca Hibernate mapovali na príslušné entity v databáza sa nachádzajú v balíku sk.stuba.fiit.ml.dp.db.structures.marc a sk.stuba.fiit.ml.dp.db.structures.marc.

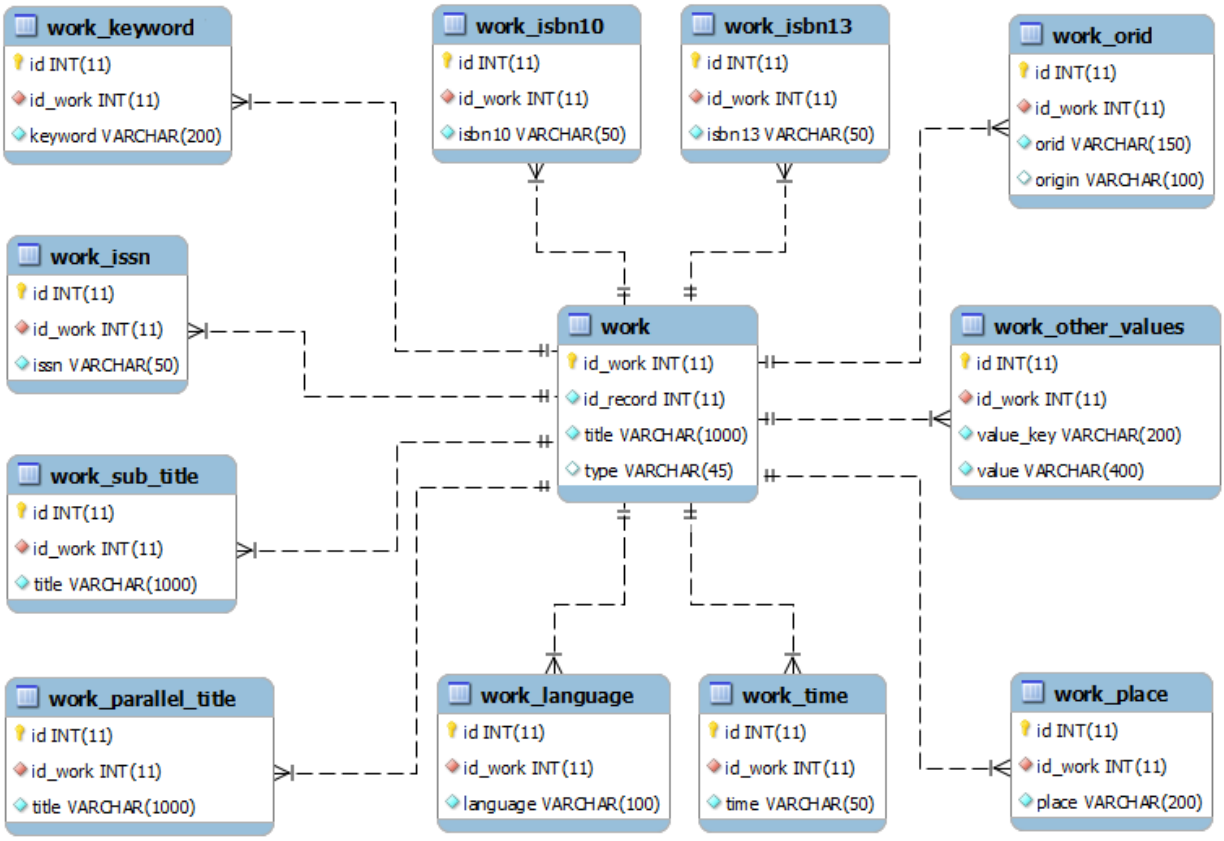
4.3.1 Fyzický model databázy M2VDB

Vysvetlivky: rimárny kľúč udzí kľúč vinné pole nepo né pole

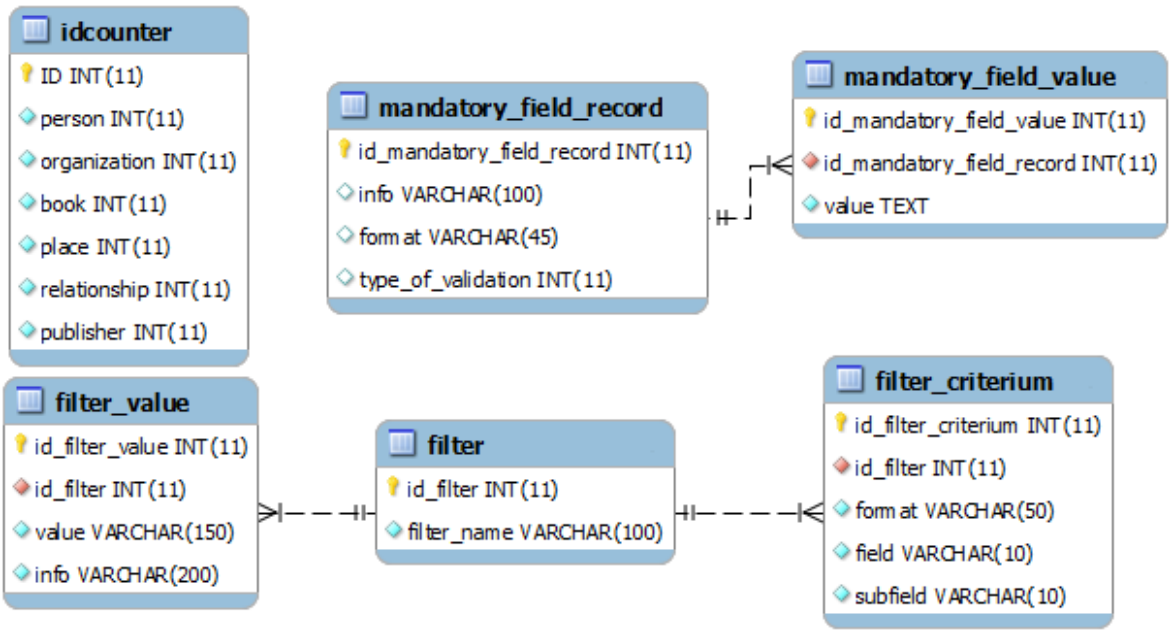


Obrázok č. 15 Fyzický model databázy M2VDB.

Entita work je kvôli prehľadnosti znázornená samostatným diagramom.



Obrázok č. 16 Entita work združuje súvisiace informácie o diele pomocou viacerých tabuliek



Obrázok č. 17 Zvyšné entity databázy.

4.4 Prevod rolí z UNIMARCu do MARC 21

Ako bolo spomenuté v návrhu, MARC21 a UNIMARC nemajú rovnako kódované polia pre rolu určujúcu zodpovednosť za dielo. V procese predspracovania roly v UNIMARC záznamoch nahrádzame ekvivalentnými rolami z formátu MARC21. Prevod uskutočňujeme v triede sk.stuba.fiit.ml.dp.util.PersonRoleValueChanger.java. Trieda obsahuje jednu metódu
`public String getRole(String unimarcValue)`, ktorá pracuje s konfiguračným súborom *<projekt>*/resources/UNIMARCtoMARC21roles.properties.
V súbore sú zapísané dvojice: $rola_{UNIMARC} = rola_{MARC21}$.

Ukážka prevodov:

```
#actor                  #adapter                #annotator
005=act                 010=adp                 020=ann
```

Vstupom metódy je kód roli v UNIMARCu. Výstupom je jej ekvivalentná hodnota používaná vo formáte MARC21.

Používateľ môže v tomto súbore definovať nové alebo editovať existujúce prevodové vzťahy medzi kódovaniami rolí MARC formátov.

4.5 Validácia záznamov

Validáciu záznamov sme implementovali triedami, ktoré dedia od rozhrania sk.stuba.fiit.ml.dp.validation.IValidator.java. Rozhranie predpisuje tri metódy:

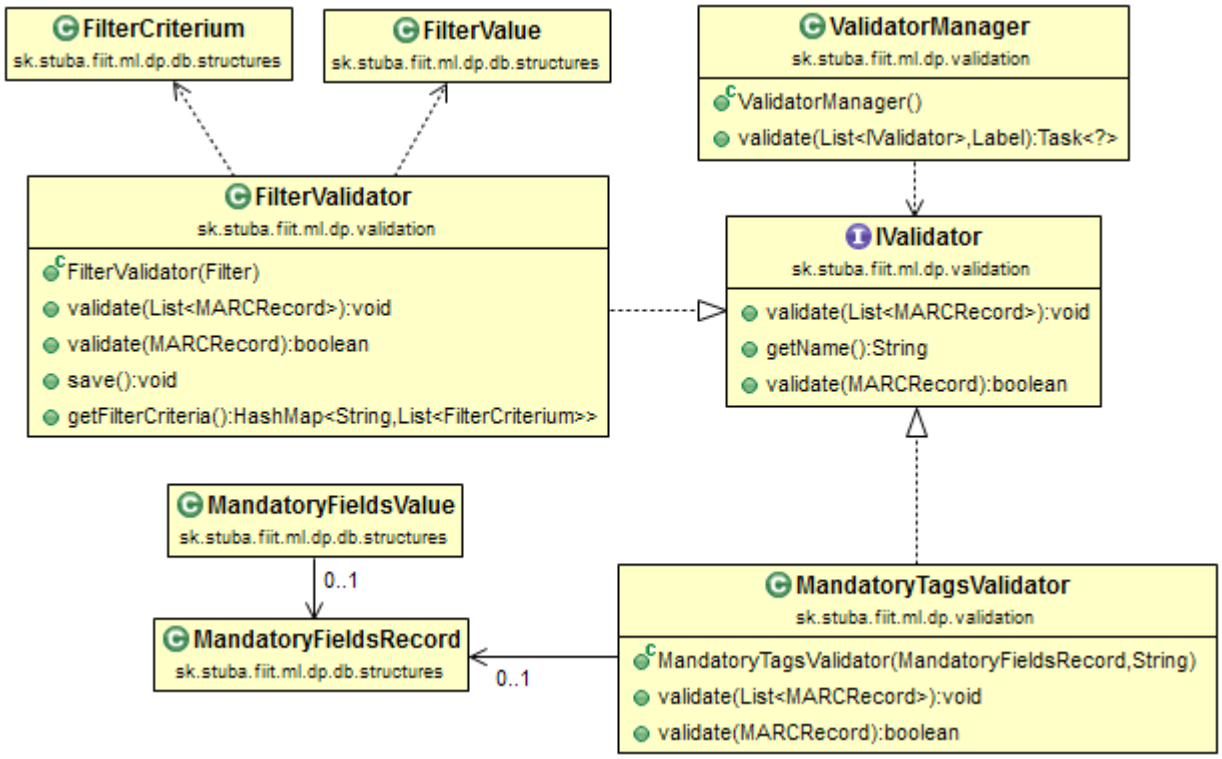
```
public interface IValidator {

    public void validate(final List<MARCRecord> recordsToValidate);
    public boolean validate(final MARCRecord marcRecords);
    public String getName();
}
```

- Prvá metóda predpisuje validáciu zadaného zoznamu bib. záznamov.
- Druhá metóda validuje jeden bib. záznam. V tele metódy musí byť implementované validačné pravidlo, ktoré aplikujeme na zvolený záznam. Výstupom funkcie je buď hodnota true, ak záznam spĺňa validačné pravidlo, alebo false v opačnom prípade.
- Tretia metóda vracia meno validačného pravidla, ktoré mu používateľ pridelil.

V súčasnosti sme sme implementovali dva typy validátorov:

MandatoryTagsValidator.java	validácia existencie povinných tagov a podpolí z bib. zázname
FilterValidator.java	validácia hodnôt v definovaných podpoliach

Oba sú dostupné prostredníctvom grafického používateľského rozhrania a je možné ich konfigurovať. Nastavenia oboch typov validátorov sa ukladajú do databázy. Hodnoty z MandatoryTagsValidator.java sú uchovávané v tabuľke mandatory_field_record a mandatory_field_value. Hodnoty z FilterValidator.java sa ukladajú do tabuľky filter, filter_criterium, filter_value. Triedy implementujúce metódy validácie sú znázornené na obrázku č. 18.



Obrázok č. 18 Diagram tried znázorňujúci štruktúru a vzťahy medzi triedami, ktoré implementujú validáciu bibliografických záznamov.

4.6 JENA - RDF

Ontologický model a jeho jednotlivé časti (triedy, vlastnosti a vzťahy) sme implementovali pomocou vývojového rámca Apache Jena[39].

V triede sk.stuba.fiit.ml.dp.jena.TripletModel.java sú implementované metódy na vytváranie, editovanie a ukladanie jednotlivých súčastí VIVO modelu.

Trieda obsahuje metódy typu:

```
private Resource createPerson(String URI, String firstName, String lastName)
private Resource createBook(String URI, String title)
private void addNumberOfPages(Resource thing, String numberOfPages)
```

Najdôležitejšiu je metóda `public void save(Record record)`, ktorá dostáva na vstupe spracovaný bibliografický záznam (identifikované triedy majú pridelený URI identifikátor). Metóda následne vytvára triplety z údajov obsiahnutých v bibliografickom zázname, deduplikuje vytvorené triplety, ak sa už nachádzajú vo VIVO modeli, do ktorého ich zapisujeme.

[39] https://jena.apache.org/

Metódy na dopytovanie sa nad VIVO model sme implementovali v triede sk.stuba.fiit.ml.dp.jena.DataModelManager.java. Metódy obsahujú zápis SPARQL dopytov, ktoré sa vykonávajú nad modelom. Ako bolo uvedené v návrhu, vyhľadávanie inštancií podľa textového názvu (napr. meno osoby alebo vydavateľa) sme vylepšili vytvorením a použitím vlastnej vyhľadávacej funkcie, ktorá umožňuje vyhľadávať textové reťazce v definovanom rozsahu podobnosti.

Funkcia dedí štruktúru triedy `FunctionBase3` z balíka `com.hp.hpl.jena.sparql.function`.

```
public class MyCustomSearchFunction extends FunctionBase3 {

public MyCustomSearchFunction() { super() ; }
    JaroWinkler jw = new JaroWinkler();

   @Override           // name1    // name2    // threshold
   public NodeValue exec(NodeValue arg0, NodeValue arg1, NodeValue arg2) {
    if (jw.getSimilarity(arg0.getString(), arg1.getString()) >= arg2.getDouble())
    return NodeValue.TRUE; // podobnosť je väčšia alebo rovná ako sme definovali
    return NodeValue.FALSE; // podobnosť slov je menšia ako sme definovali
   }
}
```

Funkcia musí mať v rámci modelu predpísaný prefix a URI adresu, s ktorou je zviazaná. URI adresa predstavuje java balíček, v ktorom sa nachádza trieda `MyCustomSearchFunction`.

```
PREFIX myfunc: <java:sk.stuba.fiit.ml.dp.jena.> // URI adresa funkcie
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX rdf:  <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

SELECT DISTINCT ?id
   WHERE {
    ?id  rdf:type foaf:Person .
    ?id  rdfs:label ?personName .
    // volanie funkcie v SPARQL dopyte, kde ?personName je meno z modelu
    //"hľadané  meno" je meno, ktoré hľadáme (je to String)
    // podobnosť = minimálna hodnota podobnosti
    FILTER(myfunc:MyCustomSearchFunction(?personName,<hľadané meno>,<podobnosť>)).
   }
```

Podobnosť je reálne číslo v rozsahu <0.0 , 1.0>, určuje minimálnu hranicu podobnosti (napr. 0,97 = 97%). Hodnoty hraníc podobnosti sa nastavujú v konfiguračnom súbore *<projekt>*/resources/thresholdconfig.properties alebo cez grafické používateľské rozhranie v časti Spracovanie →Rozpoznávanie.

4.6.1 Definovanie tried, vlastností a vzťahov

V súčasnosti sú zoznamy ontológií, triedy, vlastnosti a vzťahy medzi nimi implementované staticky. Identifikované triedy z VIVO modelu sú implementované v triede sk.stuba.fiit.ml.dp.jena.OntologyClass.java. Zoznam použitých vlastností a vzťahov sa nachádza v triede sk.stuba.fiit.ml.dp.jena.OntologyProperty.java. Použité ontológie (resp. ich názvoslovia) sa nachádzajú

v triede sk.stuba.fiit.ml.dp.jena.Ontology.java. Menný priestor (*namespace*) je definovaný v triede sk.stuba.fiit.ml.dp.jena.Namespace.java.

Pri vytváraní URI identifikátorov sme použili vlastný menný priestor `http://ml.fiit.stuba.sk/`.
Do budúcnosti plánujeme presun obsahu spomenutých tried do konfiguračných súborov alebo do databázy, čím dosiahneme väčšiu konfigurovateľnosť systému.

4.7 Transformácia bib. dát

Hlavnou triedou riadiacou proces transformácie bib. dát do ontológie je ProcessManager.java. Trieda spúšťa metódy predspracovania, ktoré sme implementovali v triede PreprocessManager.java, pre vybrané MARC formáty (Marc21.java a UniMarc.java). Vybrané údaje (resp. tagy a podpolia) sú z bib. záznamu extrahované metódami triedy Extractor.java.
Napr:

```
public List<String> extractIsbn10(List<Field> fields, String tagName)
public List<String> extractWorkParallelTitles(List<Field> fields, String tagName)
```

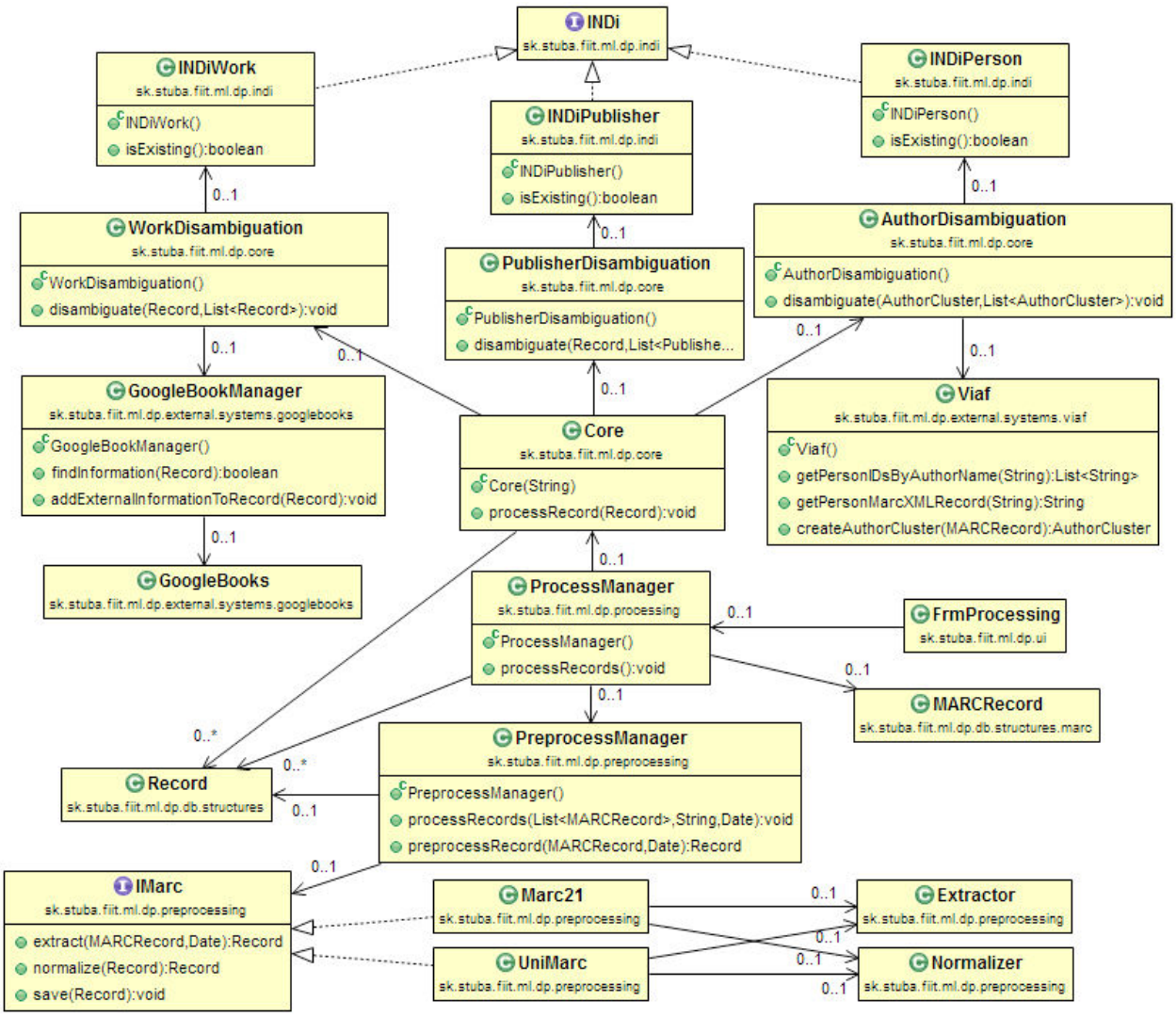
Normalizačné pravidlá sme implementovali v triede Normalizer.java. Obsahuje napríklad metódu:
`private String normalizeISBN13(String isbn13)`, ktorá odstráni zo zadaného ISBN všetky pomlčky a iný opisný text, ktorý môže byť uvedený s týmto identifikátorom v bib. zázname.

Trieda Core.java slúži ako jadro transformácie. Obsahuje metódy na predprípravu a štrukturovanie bib. záznamov a prípravu zhlukov, s ktorými budeme porovnávať daný bib. záznam.

Pripravené štruktúrované bib. záznamy a zhluky sú poskytnuté triedam, WorkDisambiguation.java, PublisherDisambiguation.java a PersonDisambiguation.java, ktoré slúžia na identifikáciu inštancií bibliografických entít. Každá z uvedených tried obsahuje metódu `disambiguate()`. Úlohou metódy je prideliť URI identifikátor vybranej inštancii. Pri tejto úlohe využívajú INDi algoritmus, ktorý sme implementovali v triedach INDiWork.java (pre diela), INDiPerson.java, (pre osoby) a INDiPublisher.java, (pre vydavateľov).

Uvedené triedy dedia od spoločného rozhrania INDi, ktoré im predpisuje metódy, ktoré musia implementovať. Každá INDi trieda obsahuje metódu `isExisting()`. Metóda dostáva na vstup inštanciu (osobu, dielo, alebo vydavateľ) z bib. záznamu a zoznam/zhluk inštancií (osoby, diela, vydavatelia) z VIVO modelu s ktorými sa daný bib. záznam porovnáva a hľadá najväčšia zhoda. V tele metódy sú implementované atribúty, ktoré sa porovnávajú (napr,: u osôb je to meno, rok narodenia, spoluautori, a iné). Výsledkom funkcie je hodnota `true` alebo `false` hovoriaca o nájdení alebo nenájdení zhody. V závislosti od výsledku sa buď generuje nové URI alebo priraďuje URI objavenej inštancie z VIVO modelu.

Triedy WorkDisambiguation.java a PersonDisambiguation.java rozhodujú o zapojení externých systémov v procese rozlišovania. Získavanie informácií z externých zdrojov sme implementovali v triedach Trieda GoogleBookManager.java a Viaf.java. Vzťahy medzi jednotlivými triedami sú znázornené diagramom tried na obrázku č. 19.



Obrázok č. 19 Diagram tried znázorňujúci jadro systému M2V. V diagrame sú zobrazené iba niektoré metódy tried.

INDi triedy čerpajú nastavenia definovaných prahových hodnôt a bodových koeficientov podobnosti z konfiguračného súboru <*projekt*>/resources/disambiguationconfig.properties.
V súbore sa nachádzajú štyri tipy hodnôt:

- hodnoty bodových koeficientov používaných pri porovnávaní parametrov

  napr: a_yearOfBirthMatch=2.0 `#hodnota koeficientu pri zhode v roku narodenia`
  w_yearMatch=0.5 `#hodnota koeficientu pri zhode v roku publikovania`
  p_nameMatch=1.5 `# hodnota koeficientu pri zhode v názve vydavateľa`

  predpona a_ sa vzťahuje na koeficienty týkajúce sa identifikácie osôb
  predpona w_ sa vzťahuje koeficienty týkajúce sa identifikácie diel
  predpona p_ je určená pre koeficienty vydavateľov

- minimálne prahové hodnoty podobnosti používané v INDi algoritme:

```
trsh_person_with_coauthors=6.5      # minimálna prahová hodnota podobnosti
                                    # pri identifikácií autora ak sa v procese
                                    # identifikácie hľadajú zhody v spoluautoroch

trsh_person_without_coauthors=5.5   # minimálna prahová hodnota podobnosti
                                    # pri identifikácií autora ak sa v procese
                                    # identifikácie nehľadajú zhody
                                    # v spoluautoroch

trsh_work=5.5                       # minimálna prahová hodnota podobnosti pri
                                    # identifikácii diela

trsh_publisher=3.0                  # minimálna prahová hodnota podobnosti pri
                                    # identifikácií vydavateľov
```

- minimálne prahové hodnoty podobnosti používané v INDi algoritme pri identifikácií osôb v systéme VIAF:

```
viaf_with_coauthors_indi_threshold=5.5      # minimálna prahová hodnota podobnosti
                                            # pri identifikácií autora ak sa v procese
                                            # identifikácie hľadajú zhody v spoluautoroch

viaf_with_no_coauthors_indi_threshold=5.5 # minimálna prahová hodnota podobnosti
                                            # pri identifikácií autora ak sa v procese
                                            # identifikácie nehľadajú zhody # v spoluautoroch
```

- hodnoty podobnosti textových reťazcov, ktoré sa používajú ako parameter funkcie JWD:

```
jaro_winkler_threshold_person_name=0.97
jaro_winkler_threshold_work_title=0.97
jaro_winkler_threshold_publisher_name=0.97
```

Použité číselné údaje majú informatívny charakter.

4.8 Neimplementované časti systému

Zoznam častí systému, ktoré sme implementovali:

- Protokol Z39.50 na získavanie dát a metadát z knižničných systémov.
- Protokol OAI PMH na zber dát z knižničných systémov.
- Možnosť výberu ontológií a spôsob mapovania polí a podpolí MARC záznamov na zvolené triedy, vlastnosti a vzťahy vybraných ontológií.
- Štatistické výsledky zo spracovania (náhradu môže v súčasnej dobe tvoriť log, ktorý vzniká pri spracovaní záznamov).
- Rozlišovanie inštancií konferencií, žurnálov, geografických miest a iných tried, ktoré sú súčasťou bibliografických záznamov.

5 Overenie riešenia

V priebehu implementácie sa uskutočnilo niekoľko menších testov. Testovacie sady obsahovali od 100 do 1000 bib. záznamov. Cieľom bolo správne nastavenie škál a bodovacích hodnôt určujúcich presnosť algoritmov na rozlišovanie inštancií a pojmov.

Vo finálnej fáze sa uskutočnilo testy nad väčším korpusom dát. Bibliografické dáta boli čerpané zo Slovenskej poľnohospodárskej univerzity v Nitre. Testovací korpus obsahoval 5751 záznamov (formát MARC 21). V tabuľke č. 6 sú uvedené štatistické údaje týkajúce sa testovacích záznamov.

Tabuľka č.6. Štatistické údaje z testovacích záznamov

Entita	Názov atribútu	Počet
Autor	Počet inštancií osôb v záznamoch	20568
	Priemerný počet spoluautorov autorov	2,58
	Skutočný počet	3651
	Rôzne mená	3575
	Rovnaké priezvisko, rovnaké meno, iný interný identifikátor	76
	Rovnaké priezvisko, rôzne meno	148
	Iniciály v mene	121
	Počet rôznych identifikátorov	3754
	Počet osôb s dátumom narodenia	437
	Počet osôb s dátumom úmrtia	7
	Počet osôb s rôznym menom (preklep, iná variantná forma)	54
Dielo	Počet v korpuse	5751
	Počet diel s rovnaký hlavným názov	395
	Počet diel s rovnaký hlavným názov a podtitulom	18
	Obsahuje paralelný názov	2683
	ISBN10, ISBN13	1207
	Rok vydania	1374
	Obsahuje vydavateľa	1320
	Priemerný počet autorov na záznam	3,58
	Maximálny počet autorov v zázname	67
	Obsahuje jazyk	5751
	Počet rôznych jazykov	24
	Skutočný počet rôznych diel	5744
Vydavateľ	Počet záznamov obsahujúce	1320
	Rôzne názvy	194
	Skutočný počet rôznych	175
	Maximálny počet vydaných diel	882
	Obsahovali miesto	1312

5.1 Predspracovanie

Nevyhnutnou podmienkou úspešného spracovania a presného rozlíšenia inštancií je predpríprava a čistenie záznamov. Záznamy boli importované do systému, kde boli validované a vybrané informácie z nich transformované/normalizované a uložené do databázy.

Validované záznamy museli obsahovať nasledujúce polia:

pole	Názov
001	Identifikátor záznamu
245 a	názov diela
100 a | 700 a	Meno osoby (hlavná a vedľajšia zodpovednosť)
100 4 | 700 4	Kód roly (autor, editor,...)
100 7 | 700 7	Identifikátor osoby v rámci knižničného systému

Polia 100 7 a 700 7 boli kontrolované na základe medzinárodného číselníka kódov roly pre MARC21.

Následne sa skontrolovala neprázdnosť podpolí (ak existovali v zázname):
001, 100 a, 100 4, 100 7, 242 a, 245 a,245 b, 246 a. 700 a, 700 4 a 700 7.

V ďalšom kroku sa uskutočnila extrakcia údajov a ich normalizácia. Extrahovali sa polia:
001, 020 a, 040 b, 041 ab, 080 a, 100 ad47, 242 a, 245 ab, 246 a, 247 a, 250 a, 260 ac, 300 a, 500 a, 650 a, 700, 856 u, 928 7 a 970 a.

Údaje boli extrahované a normalizované:

- rozdelenie mena osoby z pola 700 a na samostatné meno a priezvisko
- odstránenie znaku čiarky na konci mena osoby
- prevod ISBN kódu do tvaru bez pomlčiek a prídavného textu týkajúceho sa typu väzby diela
- odstránenie špeciálnych bib, značiek na konci názvu diela (: = / .)
- normalizácia roku vydania, odstránenie znakov [ ]
- odstránenie špeciálnych bib. znakov z názvov miest ([ ] / , . :)
- odstránenie konštánt [s.n.] a [s.l.], ktoré sa môže vyskytnúť v poli 260a a 260c.
- rozdelenie spojenej hodnoty roku narodenia a úmrtia osoby do samostatných hodnôt
- extrakcia informácie o počte a rozsahu strán diela z poľa 300. V poli sa môže nachádzať informácia o celkovom počte strán alebo číslo začiatočnej a koncovej strany.
- spojenie identifikátora diela (kontrolné pole 001) s názvom zdroja záznamov. Snahou je vytvoriť jedinečný identifikátor.

5.2 Testovanie

Cieľom testovanie bolo overenie správnosti výsledkov implementovaného INDi algoritmu v procese rozlišovania inštancií osôb, diel a vydavateľov. Zisťovali sa počty a správna identifikácia inštancií pri rôznych vstupoch koeficientov pre zvolené parametre porovnávania. Počas testov sa menil aj parameter určujúci podobnosť textových reťazcov na základe metriky Jaro-Winkler distance, čo umožnilo väčšiu škálovateľnosť pri vyhľadávaní a porovnávaní textových reťazcov ako mená alebo názvy, kde mohlo dôjsť k preklepom alebo gramatickým chybám.
Niektoré atribúty (ako napr. mená osôb, vydavateľov) boli normalizované (NT). Normalizácia spočívala v prevode na malé písmená, odstránenie počiatočných a koncových medzier a zmeny ypsilonov na jóty.

Merali sa dva hlavné ukazovatele:

- počet správne identifikovaných inštancií
- počet nesprávne identifikovaných inštancii. Nesprávne identifikovaná inštancia môže mať dve formy:
    - nesprávne zlúčenie s inou inštanciou
    - vytvorenie duplicity

Počet správne a nesprávne identifikovaných inštancií diel, autorov a vydavateľov sa určoval na základe porovnania vytvoreného VIVO modelu s referenčným modelom, ktorý vznikol deduplikáciou testovacieho korpusu.

5.2.1 TEST č.1

V prvom teste sa sledovala závislosť správnej identifikácie inštancií na základe meniacich sa hodnôt hodnotiacich parametrov. Vytvorili sme tri rôzne sady hodnôt testovacích parametrov. Každý test má inak nastavené bodovacie hodnoty a škály v rozlišovacom INDi algoritme. Pri porovnávaní textových reťazcov sa používala exaktná zhoda (reťazce sa museli rovnať).
Hodnoty koeficientov porovnávaných parametrov pre jednotlivé triedy sú uvedené v tabuľke č. 7.

Tabuľka č. 7 Hodnoty koeficientov porovnávaných parametrov pre test č.1

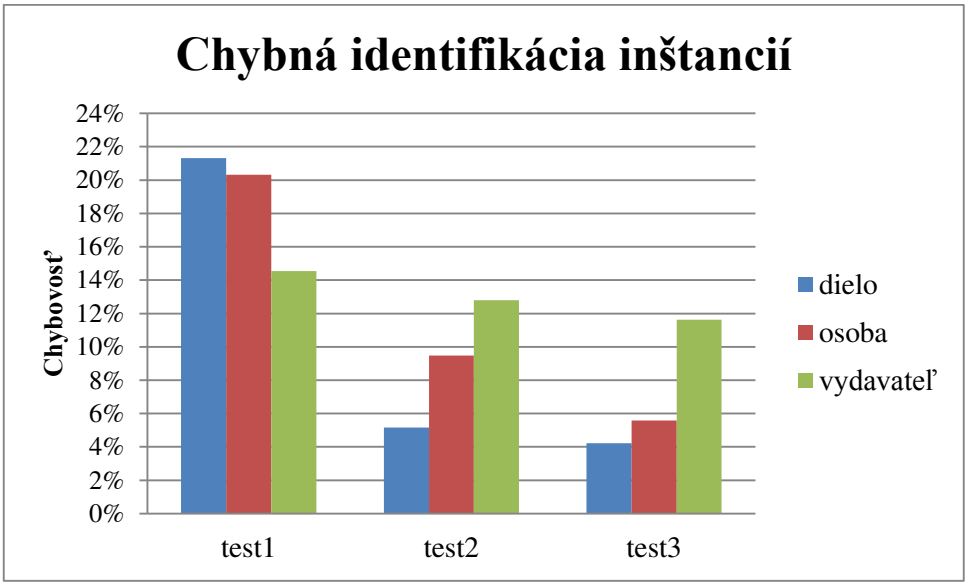
Trieda	Názov hodnotiaceho parametru	Sada 1	Sada 2	Sada 3
		Hodnoty	Hodnoty	Hodnoty
Osoba	Zhoda v priezvisku a krstnom mene	3.0	2.4	2.5
	Zhoda v priezvisku a krstnom mene (NT)	1.0	2.0	2.0
	Zhoda v priezvisku a iniciále mena	1.0	1.5	2.5
	Zhoda v priezvisku a iniciále mena (NT)	1.25	1.25	2.0
	Zhoda v roku narodenia	0.5	1.0	2.0
	Zhoda v roku úmrtia	0.5	1.0	2.0
	Zhoda v identifikátore z iného systému	1.0	2.0	2.0
	Zhoda v dielach	0.5	1.5	1.5
	Zhoda v spoluautorstve	1.75	1.75	2.75
	Jaro-Winkler distance	1.0	1.0	1.0
	Prahová hranica totožnosti	**4. 0**	**5.25**	**6.5**
	Prahová hranica totožnosti bez spoluautorov	**3.7**	**3.7**	**5.5**

Dielo	Zhoda v hlavnom názve diela	1.0	3.0	3.0
	Zhoda v hlavnom názve diela (NT)	0.25	2.25	2.25
	Zhoda v paralelnom názve diela	1.0	2.5	2.0
	Zhoda v paralelnom názve diela (NT)	0.5	2.0	2.0
	Zhoda v podnázve diela	0.5	0.5	1.0
	Zhoda v ISBN/ISSN	1.0	3.0	3.0
	Zhoda v mieste vydania	0.75	0.5	0.5
	Zhoda v roku vydania	0.75	0.5	0.5
	Zhoda vo vydavateľovi	0.25	1.5	1.5
	Zhoda v autoroch, editoroch, prekladateľoch	1.0	2.0	2.25
	Jaro-Winkler distance	1.0	1.0	1.0
	Prahová hranica totožnosti	**1.25**	**5.5**	**5.5**
Vydavateľ	Zhoda v názve vydavateľa	1.25	1.5	1.5
	Zhoda v názve vydavateľa (NT)	1.25	1.5	1.5
	Zhoda v alternatívnom názve vydavateľa	1.25	1.25	1. 5
	Zhoda v alternatívnom názve vydavateľa (NT)	1.0	1.25	1. 5
	Zhoda v identifikátore z iného systému	1.5	1.5	1.5
	Zhoda v dielach	0.25	1.25	1.25
	Zhoda v spolupracujúcich autoroch	0.25	0.25	0.25
	Jaro-Winkler distance	1.0	1.0	1.0
	Prahová hranica totožnosti	**1.5**	**3.25**	**3.0**

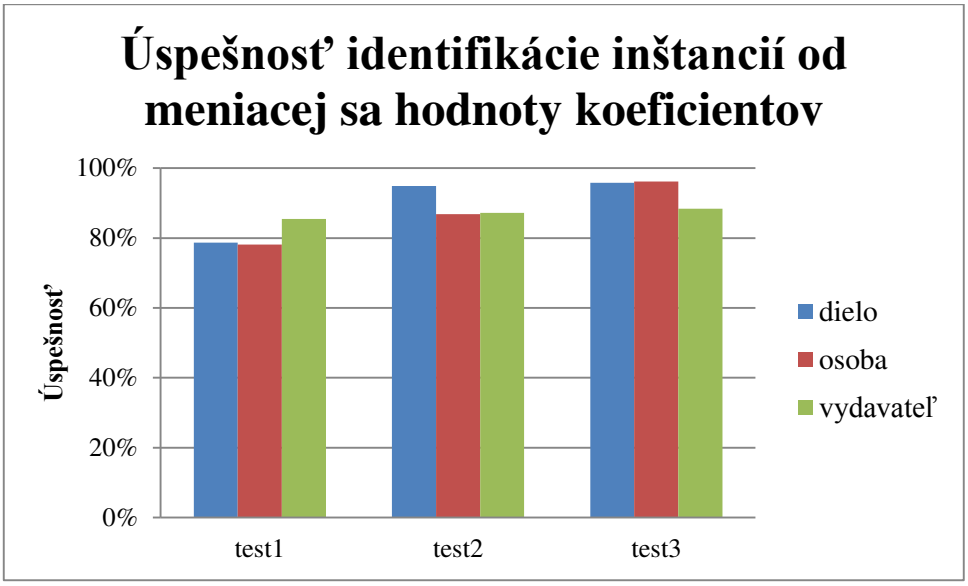
Výsledky

Tabuľka č. 8 Výsledky prvého testu

Test	**Test č.1**			**Test č.2**			**Test č.3**		
Trieda	**Dielo**	**Osoba**	**Vydavateľ**	**Dielo**	**Osoba**	**Vydavateľ**	**Dielo**	**Osoba**	**Vydavateľ**
Správne identifikované	4520	2851	147	5497	2909	150	5502	3531	152
Nesprávne identifikované	1224	742	23	247	606	24	242	204	20
Duplicity	0	580	25	21	248	22	23	168	20
Nesprávne zlúčenie	1224	162	0	275	98	0	219	36	0



Graf č. 1 Percentuálne výsledky nesprávnej identifikácie inštancií v závislosti od zmeny koeficientov porovnávania.



Graf č. 2 Percentuálne výsledky úspešnosti identifikácie inštancií v závislosti od zmeny koeficientov porovnávania.

Zhodnotenie

Prvý testovací prípad dosiahol úspešnosť pod 80%. Príčinou bol zlý odhad prahových hodnôt podobnosti a stanovenie nízkych hodnôt koeficientov. Postupným zvyšovaním hodnôt v ďalších testovacích prípadoch sa nám podarilo dostať na hranicu 90% úspešnosti. Chybovosť pri identifikácii osôb pomohlo znížiť zavedenie generovania variantných foriem mien ako aj iniciály, ktoré sa ukladajú do VIVO modelu. V druhom a treťom prípade zvýšili váhu (koeficient) pre daný parameter.

5.2.2 TEST č. 2

V druhom teste sme na základe analýzy výsledkov prvého testu vybrali najlepší výsledok a upravovali sme hodnoty metriky Jaro-Winkler distance s cieľom nájsť optimálnu hodnotu podobnosti, pri porovnávaní textových reťazcov v uvedenom korpuse.
Použité testovacie hodnoty metriky Jaro-Winkler distance boli 0.90, 0.95, 0.97 a 1.0.

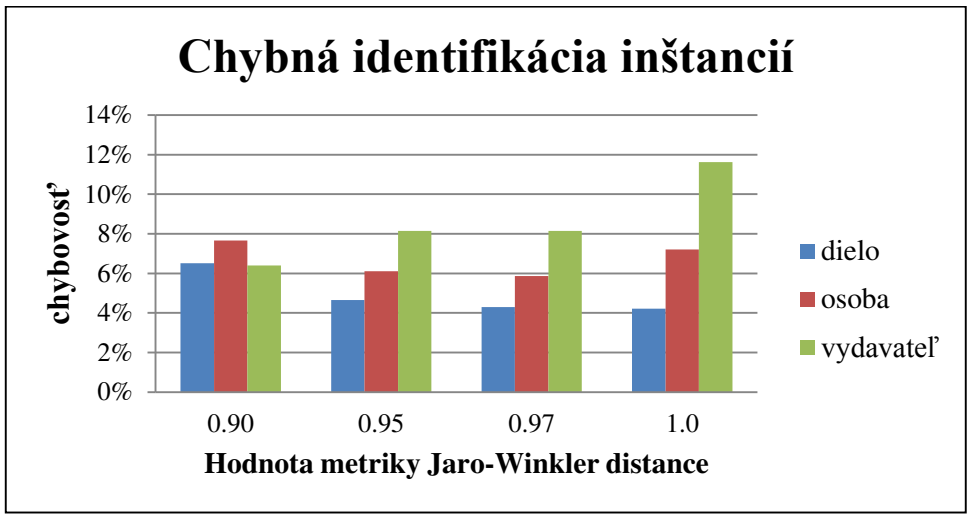
Tabuľka č.9. Hodnoty vstupných parametrov v INDI algoritme

Trieda	Názov hodnotiaceho parametru	Hodnota
Osoba	Zhoda v priezvisku a krstnom mene	2.5
	Zhoda v priezvisku a krstnom mene (NT)	2.0
	Zhoda v priezvisku a iniciále mena	2.5
	Zhoda v priezvisku a iniciále mena (NT)	2.0
	Zhoda v roku narodenia	2.0
	Zhoda v roku úmrtia	2.0
	Zhoda v identifikátore z iného systému	2.0
	Zhoda v dielach	1.5
	Zhoda v spoluautorstve	2.75
	Prahová hranica totožnosti	6.5
	Prahová hranica totožnosti bez spoluautorov	**5.5**
Dielo	Zhoda v hlavnom názve diela	3.0
	Zhoda v hlavnom názve diela (NT)	2.25
	Zhoda v paralelnom názve diela	2.0
	Zhoda v paralelnom názve diela (NT)	2.0
	Zhoda v podnázve diela	1.0
	Zhoda v ISBN/ISSN	3.0
	Zhoda v mieste vydania	0.5
	Zhoda v roku vydania	0.5
	Zhoda vo vydavateľovi	1.5
	Zhoda v autoroch, editoroch, prekladateľoch	2.25
	Prahová hranica totožnosti	**5.5**
Vydavateľ	Zhoda v názve vydavateľa	1.5
	Zhoda v názve vydavateľa (NT)	1.5
	Zhoda v alternatívnom názve vydavateľa	1. 5
	Zhoda v alternatívnom názve vydavateľa (NT)	1. 5
	Zhoda v identifikátore z iného systému	1.5
	Zhoda v dielach	1.25
	Zhoda v spolupracujúcich autoroch	0.25
	Prahová hranica totožnosti	**3.0**

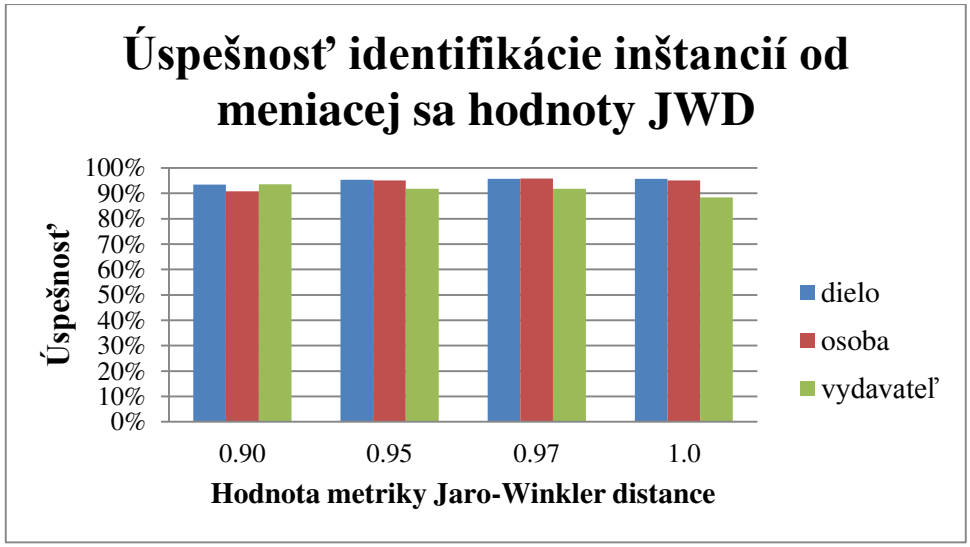
Výsledky

Tabuľka č. 10 Výsledky druhého testu

Trieda	Dielo				Osoba				Vydavateľ			
JW distance	0.90	0.95	0.97	1.00	0.90	0.95	0.97	1.00	0.90	0.95	0.97	1.00
Správne identifikované	5370	5477	5497	5502	3314	3490	3571	3531	161	158	158	152
Nesprávne identifikované	374	267	247	242	280	203	144	204	11	14	14	20
Duplicity	16	20	21	23	23	56	78	168	7	12	14	20
Nesprávne zlúčenie	358	247	226	219	257	147	66	36	4	2	0	0



Graf č. 3 Percentuálne výsledky celkovej nesprávnej identifikácie inštancií v závislosti od zmeny hodnoty metriky JWD.



Graf č. 4 Percentuálne výsledky úspešnosti identifikácie inštancií v závislosti od zmeny hodnoty metriky JWD.

Zhodnotenie

Z výsledkov vyplýva, že zmena hodnoty metriky JWD ovplyvňuje výrazným spôsobom úspešnosť identifikácie inštancií. Najlepšie výsledky pri identifikácií osôb a diel sa dosiahli pri hodnote 0.97, teda pri porovnávaní textových reťazcov s hodnotou podobnosti nad 97%. Táto hranica porovnania pomáha prekonávať problémy pri porovnávaní názvov a mien, v ktorých sa nachádza menšia gramatická chyba alebo preklep. Vďaka metrike JWD boli v testovacích dátach úspešne identifikované prípady napr.: Vydavateľstvo EKONÓM, Vydavateľstvo EKONÓM (zdvojená medzera v názve) alebo cudzokrajné názvy, v ktorých pri prepise do slovenského jazyka môžu vzniknúť chyby resp. variantné formy mena: Grigorjeva, O. V. a Grigorieva, O. alebo Bulgakov B.M. a Bulgakov V.M. (prepis ruského písmena V na B).

Použitie príliš malej hodnoty podobnosti pri porovnávaní mien osôb a názvov diel vedie k zvýšeniu chybovosti v procese identifikácie. Veľká voľnosť pri porovnávaní mien osôb vedie k zlučovaniu rozdielnych mien, napr.: Angelovič, Marek a Angelovič, Michal, ktorých podobnosť je 0.9436275.

Výsledky testu potvrdzujú správnosť výberu metriky Jaro-Winkler distance, použitej pri porovnávaní a hľadaní zhody v textových reťazcoch. Ako bolo spomenuté v analýze, táto metrika je vhodná pre jazyky, v ktorých sa využíva skloňovanie alebo časovanie resp. zmena prípony slova. Túto vlastnosť sme využili pri hľadaní zhôd v názvoch organizácií, v ktorých sa nachádza názov mesta alebo označenie typu organizácie napr:

Ústav vedecko-technických informácií pre poľnohospodárstvo v Nitre
Ústav vedecko-technických informácií pre pôdohospodárstvo.

Podobnosť daných názvov je 0.9652666.

Alebo Agroinštitút
Agroinštitút, š.p.
Podobnosť názvov je 0.95555556.

V určitých prípadoch sa môže stať, že metrika neidentifikuje všetky podobné inštancie. Jedná sa najmä o prípady, kedy má inštancia viac variantných foriem názvu alebo mena, ktoré sú dostatočné odlišné. Riešením by mohlo byť zníženie hranice podobnosti JWD. Ako sa ukázalo v testoch, zníženie hranice podobnosti spôsobilo zníženie množstva duplicít, no na druhej strane, výrazným spôsobom vzrástol počet nesprávne zlúčených inštancií a celkový nárast chybovosti.

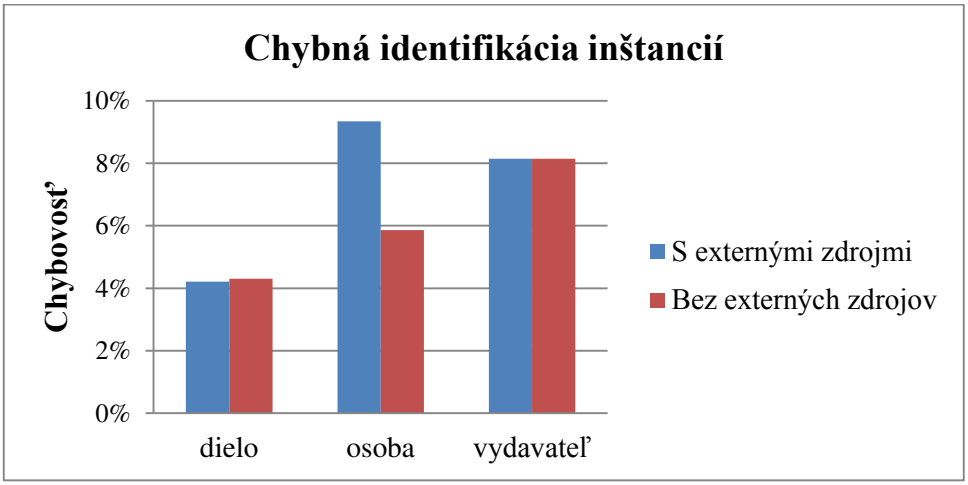
5.2.3 TEST č. 3

V treťom teste boli využité informácie z externých zdrojov VIAF a GoogleBooks. Testy boli spustené s rovnakými parametrami ako v kapitole 1.2.3 TEST č.2. Vyhľadávanie v databáze VIAF bolo uskutočňované zadaním mena hľadaného autora. V prípade výskytu viacerých autorov s rovnakým alebo podobným menom vo VIAFe sa použil INDi algoritmus na nájdenie najpravdepodobnejšieho autora. Bodové hranice podobnosti autora (s alebo bez spoluautorov) boli stanovené na hodnotu 5,5. Hľadaný autor vo VIAFe sa musel okrem mena zhodovať aj v iných parametroch, napr. spoluautori, rok narodenia, ISBN alebo názvy diel, ktoré publikoval.

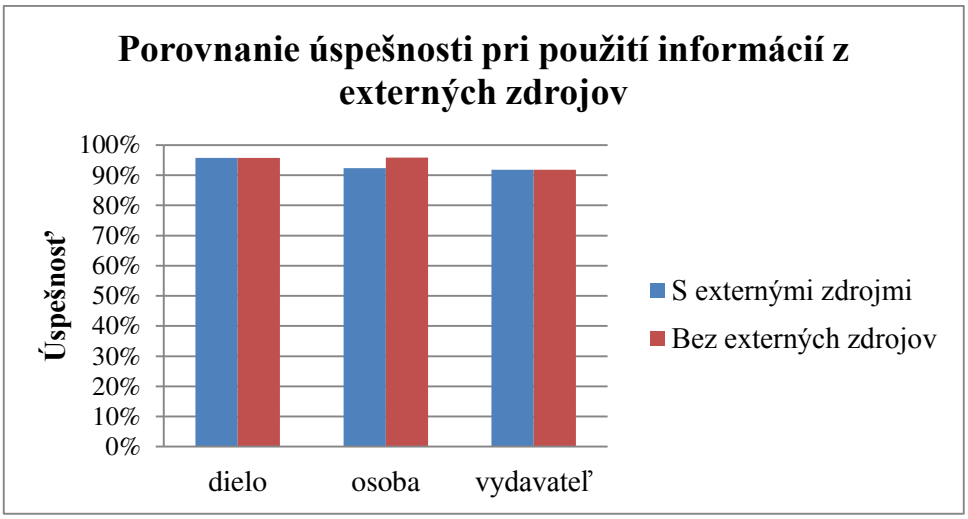
Výsledky

Tabuľka č. 11 Porovnanie výsledkov identifikácie inštancií bez a s použitím informácií z externých zdrojov.

Test	Výsledky bez použitia externých zdrojov informácií			Výsledky s použitím externých zdrojov informácií		
Trieda	**Dielo**	**Osoba**	**Vydavateľ**	**Dielo**	**Osoba**	**Vydavateľ**
Správne identifikované	5497	3571	158	5502	3373	158
Nesprávne identifikované	247	144	14	242	341	14
Duplicity	21	78	14	20	22	14
Nesprávne zlúčenie	226	66	0	222	319	0



Graf č. 5 Percentuálne porovnanie chybovosti pri použití externých zdrojov informácií v procese identifikácie inštancií



Graf č. 6 Porovnanie úspešnosti identifikácie inštancií s použitím externých zdrojov informácií VIAF a GoogleBooks.

Zhodnotenie
Z výsledkov vyplývajú tri fakty:

1) Došlo k miernemu zlepšeniu v procese identifikácie diel. Obohacujúce informácie z GoogleBooks pomohli znížiť počet nesprávnych spojení diel. V procese identifikácie sa podarilo prideliť GoogleBooks identifikátor iba 49 dielam. Spracovávané bibliografické záznamy obsahovali prevažne informácie o akademických článkoch, ktoré sa nenachádzali v databáze GoogleBooks. Prevažná časť objavených diel boli knihy, nie akademické články.
2) Získané informácie z databázy VIAF spôsobili pokles úspešnosti identifikácie inštancie osoby. Vo VIAFe sa nenachádza veľa autorov zo Slovenska. Ako bolo spomenuté, Slovensko nemá Národný súbor autorít. Slovenskí autori sa dostali do databázy VIAFu prostredníctvom zahraničných knižníc. Záznamy o slovenských autoroch obsahujú málo informácií (malé množstvo diel, spoluautorov, roky spojené s autorom, atď.). Celkovo sa podarilo identifikovať iba 338 autor, ktorí majú VIAF identifikátor. Toto číslo predstavuje 9.26% z celkového počtu autorov v spracovanom korpuse. Vyhodnocovanie potenciálnych autorov z VIAFu bolo uskutočnené implementovaným INDi algoritmom. V prípade, ak INDi algoritmus identifikoval chybne autora z VIAFu, tak sa informácie o ňom preniesli do ďalšieho porovnávania a vyhodnocovania s autormi zo spracovávaného testovacieho korpusu, čo malo za následok nabaľovanie chýb v identifikácií a celkový nárast nesprávne zlúčených autorov.
3) Pri identifikácií vydavateľov v súčasnosti nepoužívame žiadny zdroj externých informácií. Výsledky neovplyvnila ani zvýšený počet nesprávne identifikovaných autorov.

5.3 Zhrnutie testov

Testy preukázali, že použitá metóda INDi, určená na rozlišovanie inštancií osôb v na základe informácií v digitálnych knižniciach, je použiteľná a prináša zaujímavé výsledky. Testy boli vykonané na dátach z knižnice Slovenskej poľnohospodárskej univerzity v Nitre. Úspešnosť metódy sme testovali s viacerými nastaveniami koeficientov podobnosti.

Rozšírením metódy pomocou metriky Jaro-Winkler distance sa nám podarilo znížiť počet duplicít v záznamoch, v ktorých sa nachádzali inštancie obsahujúce preklep v názve alebo mene, prípadne variantnú formu názvu. Metóda dosahovala najlepšie výsledky pri použití hodnoty JWD okolo 0.97. Znižovaním tejto hodnoty začala vzrastať chybovosť identifikácie osôb a diel v dôsledku nesprávneho zlučovania s inými inštanciami. Identifikácie inštancií vydavateľov bola najúspešnejšia pri hodnote JWD okolo 0.9.

Môžeme prehlásiť, že presnosť použitej metódy je závislá na presne stanovených hraniciach podobností, ktoré závisia od kvality a podobnosti porovnávaných inštancií a korektnosti existujúceho VIVO modelu, do ktorého dané inštancie zaraďujeme. Je preto nevyhnuté najprv zanalyzovať bibliografické záznamy, ktoré chceme touto metódou spracovať.

5.4 Potenciálne zlepšenie

Metóda však nedosahovala stopercentné výsledky, z časti to zapríčinila kvalita dát a z druhej časti nezaradenie niektorých atribútov do procesu identifikácie. Jedným z týchto parametrov je poradové číslo vydania diela. V niektorých prípadoch je to jediný rozlišujúci prvok dvoch diel.

Ďalším zlepšením, ktoré by bolo potrebné overiť je modifikácia INDi algoritmu, ktorý by pri objavení nezhody niektorých stanovených atribútov (napr.: rok vydania, ISBN, podnázov a i.) udeľoval záporný koeficient, teda by dokázal znížiť dosiahnutú hodnotu podobnosti práve identifikovanej inštancie. Takéto "pokutovacie" pravidlá by sa dali aplikovať na všetky typy inštancií.

6 Zhodnotenie

V práci sme analyzovali možnosti sémantizácie bibliografických dát. Sémantizácia nám poskytuje nové možnosti reprezentácie, vyhľadávania a odhaľovania nových informácií, ktoré nemuseli byť v pôvodnej reprezentácií dostupné.

Navrhli sme automatický systém na transformáciu bibliografických dát, ktoré sa prevádzajú do modelu VIVO ontológie, určenej pre oblasť vedy a výskumu. VIVO neponúka iba rozsiahly model ontológie, ale poskytuje aj nástroje na vytváranie lokálnej a medzinárodnej siete výskumníkov.

Rozhodli sme sa naplniť vybranú časť VIVO modelu údajmi, čerpanými z bibliografických MARC záznamov. V rámci sémantizácie bibliografických záznamov sme sa zamerali na možnosti riešenia problému rozoznávania inštancií autorov v bibliografických záznamoch (*name disambiguation problem*). Na riešenie sme použili metódu INDi, ktorá bola zverejnená v októbri 2011. Metódu sme čiastočne upravili a jej následným zovšeobecnením sme ju aplikovali aj na rozlišovanie inštancií diel a vydavateľov.

Do navrhnutých procesoch rozlišovania inštancií sme zakomponovali metriku Jaro-Winkler distance, ktorá nám pomáha porovnávať a vyhľadávať textové reťazce (mená osôb, názvy diel a organizácií), v ktorých sa môže vyskytovať preklep alebo gramatická chyba. Metrika dokonca umožňuje vyhľadanie inštancií, ktoré majú podobné variabilné formy názvu. Napr: Slovenská poľnohospodárska univerzita a Slovenská poľnohospodárska univerzita v Nitre.

Implementovaný systém využíva pri rozlišovaní inštancií osôb Medzinárodný súbor autorít VIAF, vďaka ktorému prepájame naše dáta s bázou dát Linked Data. V prípade nejednoznačnosti, pri rozlišovaní inštancií diel, sa využívajú dostupné informácie z portálu GoogleBooks.

Navrhnutý systém sme implementovali a otestovali jeho funkčnosť na reálnych dátach z knižničných systémov. Vykonané testy dokázali, že nami navrhnutý systém je funkčný a umožňuje sémantizáciu bibliografických dát, rozlišovanie inštancii osôb, diel, vydavateľov a napĺňanie vybraného VIVO modelu.

6.1 Možnosti ďalšieho výskumu

Aktuálne napĺňaný model VIVO ontológie, ktorý sme si v našom systéme zvolili, obsahuje tri triedy: dielo, osoba a vydavateľ. Bolo by vhodné doplniť tento model o ďalšie triedy VIVO modelu, ako napríklad: konferencia, stretnutie, výstava, univerzita, žurnál, korporácia a ďalšie. V bibliografických záznamoch sa nachádza množstvo sémantických informácií, ktoré sme našimi tromi vybranými triedami nepokryli. Pridaním ďalších tried do VIVO ontológie sa do modelu dostane množstvo nových atribútov a informácií, ktoré môžeme použiť v existujúcom INDi algoritme na identifikáciu osôb, diel a vydavateľov.

Ďalším možným pokračovaním by mohla byť úprava použitého INDi algoritmu. Úprava metódy by spočívala v pridaní záporného ohodnotenia (trestu/pokuty), v prípade, keď budú mať porovnávané inštancie rôzne hodnoty v kľúčových poliach, napr.: rozdielne ISBN, ISSN, poradové číslo vydania, počet strán v diele, alebo rozdielny rok narodenia alebo úmrtia osoby. Navrhnutá metóda by dokázala automaticky nielen zvyšovať ale aj znižovať dosiahnutú hodnotu podobnosti (v tomto prípade už aj rozdielnosti) dvoch inštancií.

Ďalšou možnosťou ako pokračovať v odhaľovaní skrytých vzťahov v digitálnych knižniciach, by bolo zaraďovanie autorov do výskumných oblasti s inými autormi na základe kľúčových slov, ktoré sa nachádzajú v bibliografických záznamoch. Zaraďovanie by sa uskutočňovalo pomocou tezaurov a iných vhodných slovníkov.

7 Použitá literatúra

1. **COYLE, K., HILLMANN, D.** Resource description and access : cataloging rules for the 20th century. [Online] [Citované: 4. 10. 2013] http://www.dlib.org/dlib/january07/coyle/01coyle.html.

2. **CHENG, S.** *Linked Open Data for Countway Library Final report for Phase 1 (June-Nov. 2012).* 2012.

3. **DIMIĆ SURLA, B., SEGEDINAC, M., IVANOVIĆ, D.** *A BIBO ontology extension for evaluation of scientific research results.* s.l. : In Proceedings of the Fifth Balkan Conference in Informatics (BCI '12). ACM, New York, NY, USA, 2012. s. 275-278.

4. **KRAFFT, D. B., CAPPADONA, N. A.,CARUSO, B.,CORSON-RIKERT, J., DEVARE, M., LOWE, B. J., VIVO Collaboration.** *VIVO: Enabling National Networking of Scientists.* s.l. : Proceedings of the WebSci10: Extending the Frontiers of Society On-Line 26-27th, Raleigh, NC: US, 2010.

5. **PERONI, S., SHOTTON, D.** *FaBiO and CiTO: Ontologies for describing bibliographic resources and citations, Web Semantics.* s.l. : Science, Services and Agents on the World Wide Web, Volume 17, December 2012. s. 33-43.

6. **MILLER, E., OGBUJI, U., MUELLER, V., MACDOUGALL, K.** *Bibliographic Framework as a Web of Data: Linked Data Model and Supporting Services.* Washington, DC : Library of Congress, November 21, 2012.

7. **SHOTTON, D.,.** CiTO, the Citation Typing Ontology. Stockholm, Sweden : Proceedings of the Bio-Ontologies Special Interest Group Meeting 2009: Knowledge in Biology, 2009.

8. **MITCHELL, S., a kol.** *The VIVO Ontology: Enabling Networking of Scientists.* Koblenz, Germany : Proceedings of the ACM WebSci'11, 2011.

9. **IFLA.** *IFLA/UNESCO Manifesto for Digital Libraries.* s.l. : International Federation of Library, International Federation of Library, 2010.

10. **KATUŠČÁK, D.** *MARC 21 Formát pre bibliografické údaje.* Martin : Slovenská národná knižnica, 2003.

11. **VIVO.** *VIVO Ontology.* [Online] [Citované: 3. 3. 2013] https://wiki.duraspace.org/display/VIVO/VIVO+Main+Page.

12. **Ontology, The Bibliographic.** *Bibliographic Ontology Specification.* [Online] [Citované: 4. 4. 2013] http://bibliontology.com/specification.

13. **BIBFRAME.ORG.** *Model Overview.* [Online] [Citované: 5. 4. 2013]. http://bibframe.org/.

14. **de Carvalho, A.P., Ferreira, A.A., Laender, A.H.F., Gonçalves, M.A.** Incremental Unsupervised Name Disambiguation in Cleaned Digital Libraries. *Journal of Information and Data Management.* 2011, Vol. 2, 3.

15. The Bibliographic Ontology. *Bibliographic Ontology Specification.* [Online] [Citované: 4. 4. 2013.] http://bibliontology.com/specification.

16. *Social Network Analysis on Name Disambiguation and More.* **BYUNG-WON, ON.** Busan : Convergence and Hybrid Information Technology, 2008. ICCIT '08. Third International Conference on, 2008 . 978-0-7695-3407-7 .

17. **KUŠŤÁROVÁ, T.** *Miery podobnosti reťazcov.* Bratislava : Fakulta matematiky, fyziky a informatiky Univerzity Komenského v Bratislave, 2008.

18. Ariadne. *Z39.50 for All.* [Online] [Citované: 8. 11. 2013.] Dostupné na: http://www.ariadne.ac.uk/issue21/z3950/.

19. **TENNANT, R.** MARC Must die. *Publishers Weekly.* [Online] 15. 10. 2002. [Citované: 11. 4. 2013] http://libraryjournal.reviewsnews.com/index.asp?layout=article&articleid=CA250046.

20. VIVO model. *wiki duraspace.* [Online] [Citované: 3.11. 2013] https://wiki.duraspace.org/display/VIVO/VIVO+Ontology+Classes+and+Properties.v1.4.1.

21. **STYLES, R., AYERS, D., SHABIR, N.** *Semantic MARC, MARC21 and the Semantic Web.* 2008.

22. British Library Data Model - Book. *The British Library.* [Online] [Citované: 11. 9. 2013] http://www.bl.uk/bibliographic/pdfs/bldatamodelbook.pdf.

23. Metadata Service. *The British Library.* [Online] [Citované: 15. 10. 2013] http://www.bl.uk/bibliographic/datafree.html.

24. MARiMbA: conversión de MARC 21 a RDF. *Biblioteca Nacional de España.* [Online] [Citované: 11. 4. 2011] http://www.bne.es/en/Inicio/Perfiles/Bibliotecarios/DatosEnlazados/Tecnologia/Conversion.html.

25. MARiMbA. *Ontology Engineering Group.* [Online] [Citované: 11. 5, 2013] http://mayor2.dia.fi.upm.es/oeg-upm/index.php/en/downloads/228-marimba.

26. MARC/MODS RDFizer. *SIMILE.* [Online] [Citované: 27. 10. 2013] http://simile.mit.edu/wiki/MARC/MODS_RDFizer.

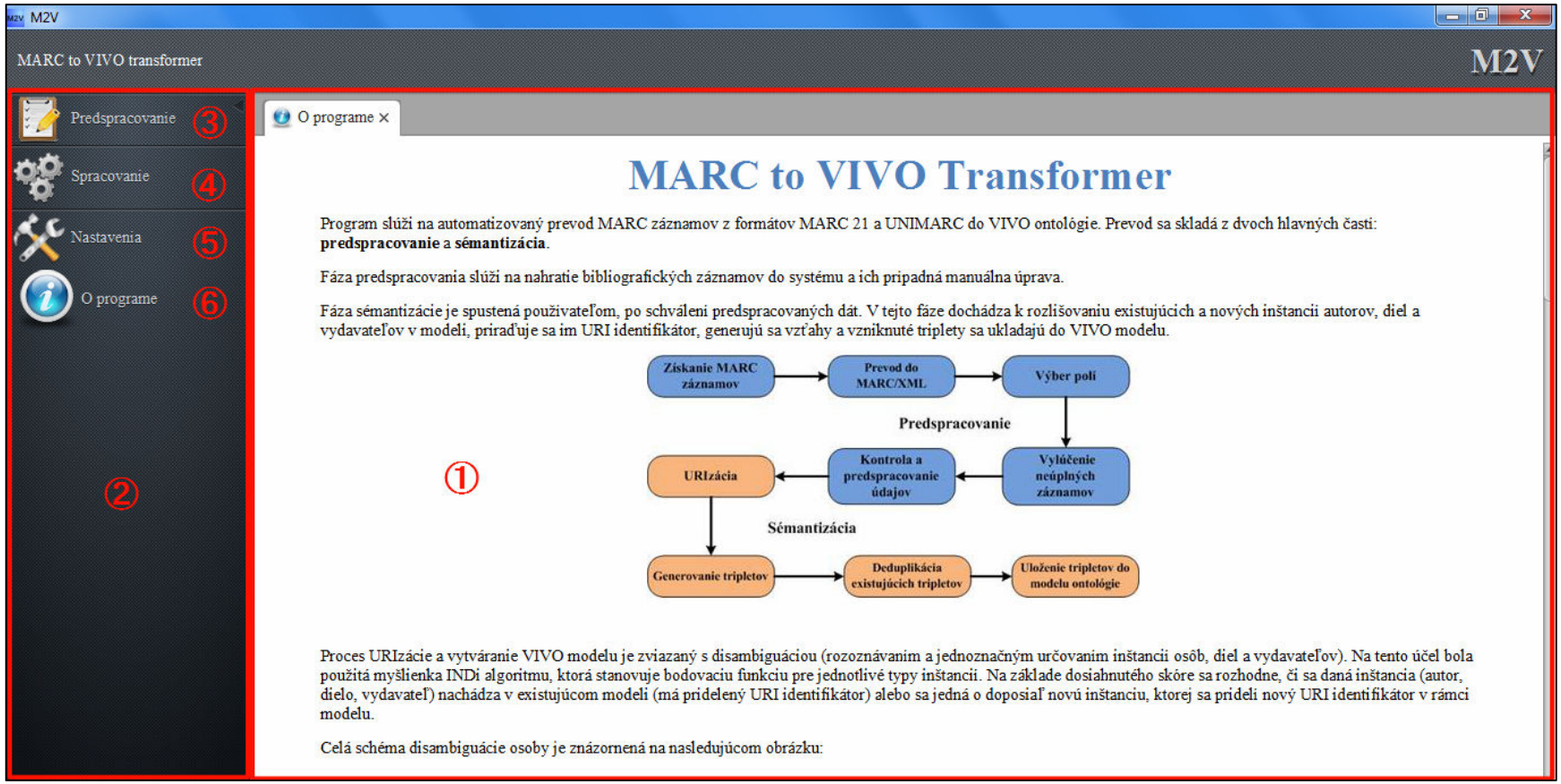
8 Technická dokumentácia

A - Zoznam obrázkov a tabuliek

A.1 Zoznam obrázkov

A.2 Zoznam tabuliek

B - Používateľské rozhranie



① Hlavné pracovné okno.

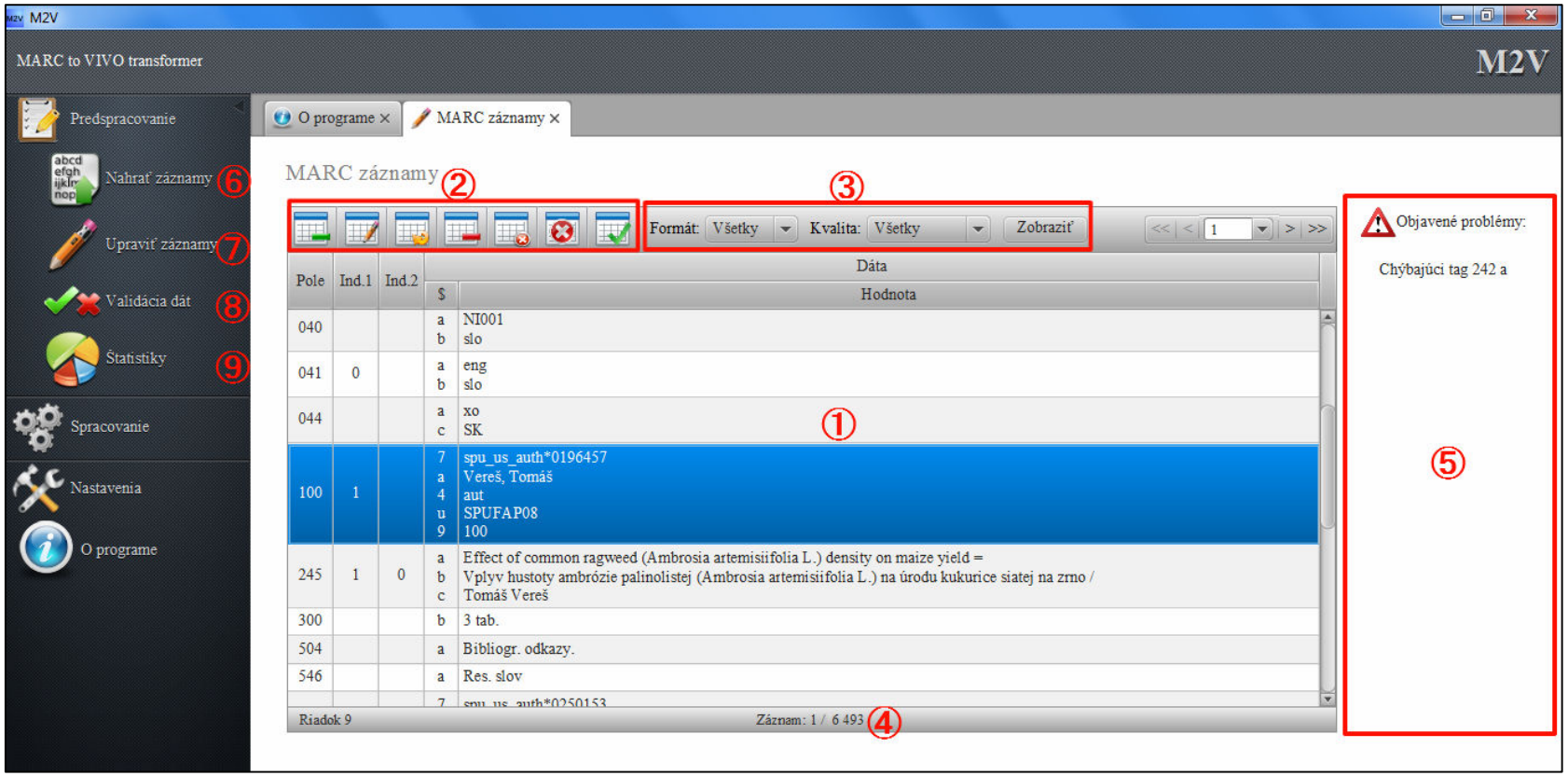
② Bočné používateľské menu, odkiaľ sú dostupné všetky funkcie systému.

③ Ponuka možností na predspracovanie dát.

④ Ponuka možností na spracovanie dát.

⑤ Ponuka možností, týkajúcich sa nastavenia systému.

⑥ Informácie o systéme.



① Zobrazenie bib. záznamov (MARC formát).

② Funkcie na manipuláciu so záznamami (pridávanie nových polí, editácia hodnôt, mazanie vybraných polí a celých záznamov, a iné).

③ Filter umožňujúci výber bib. záznamov podľa MARC formátu a kvality záznamu (všetky, validné, nevalidné nevalidované)

④ Ukazovateľ počtu záznamov.

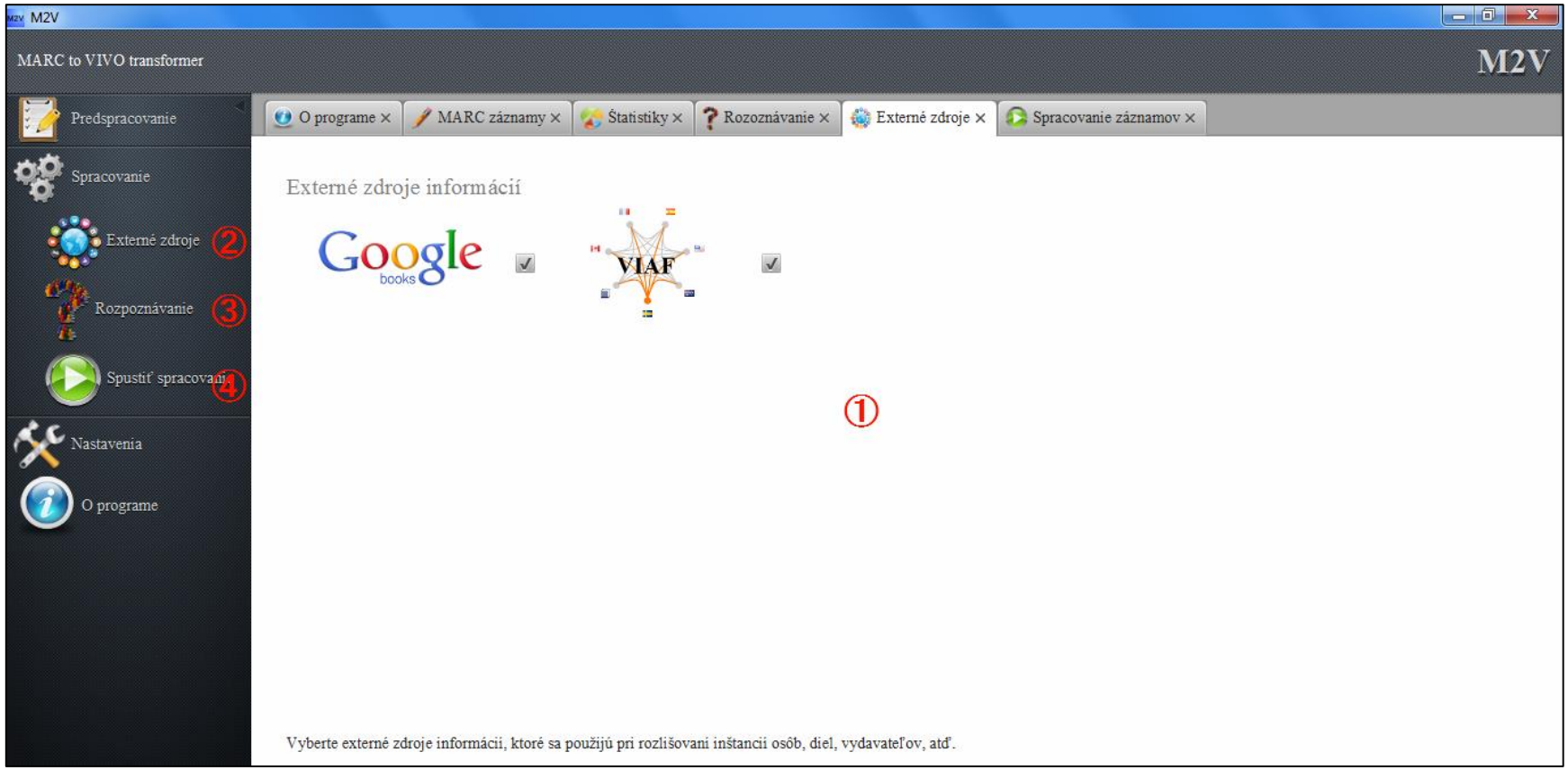
⑤ Zoznam chýb, ktoré boli detegované aplikáciou validačných pravidiel.

⑥ Ponuka na nahrávanie záznamov do systému.

⑦ Ponuka na editovanie záznamov (aktuálne okno).

⑧ Ponuka na spustenie validácie záznamov.

⑨ Štatistické informácie o bib. záznamoch v systéme.

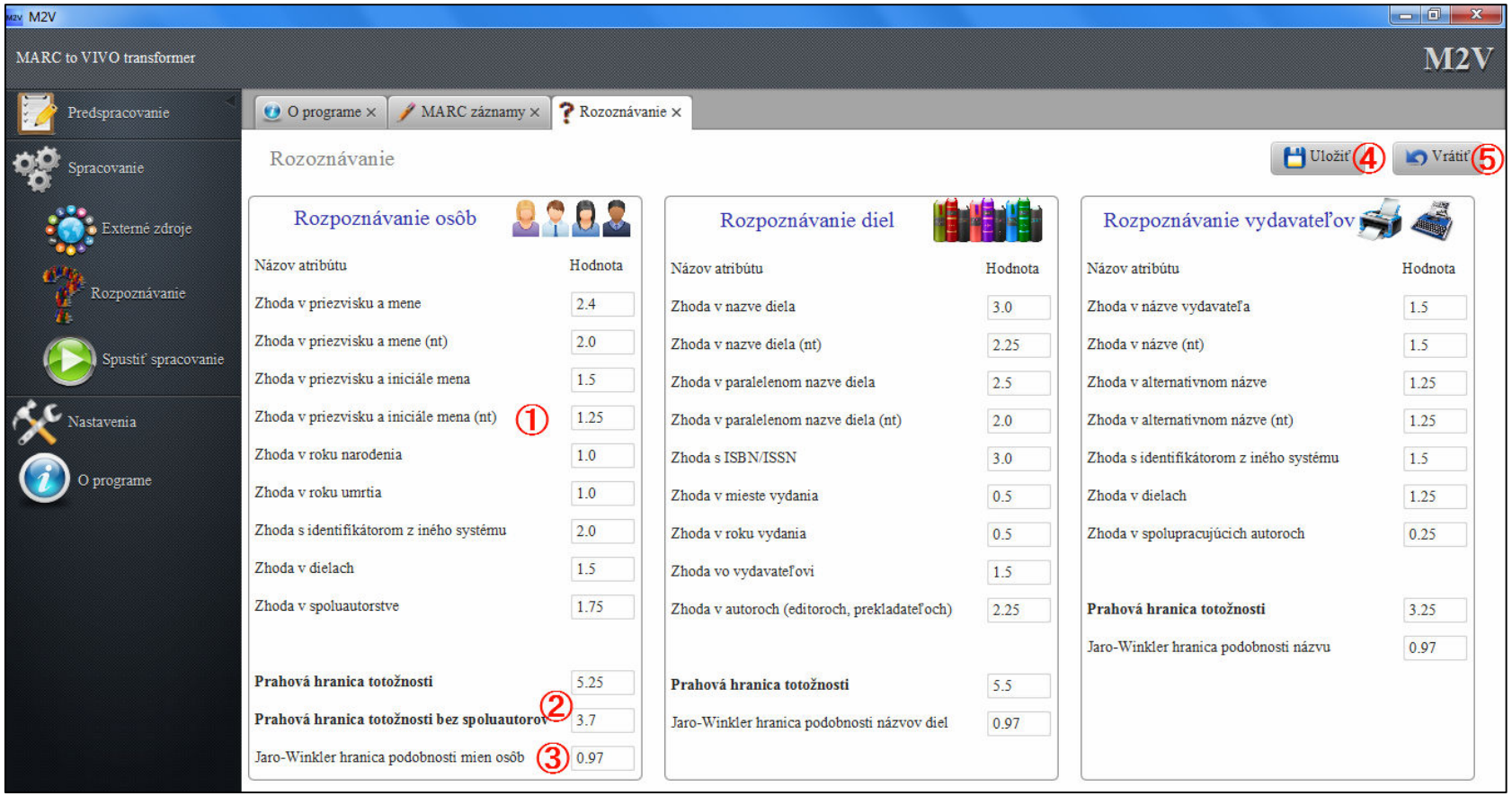


① Zoznam dostupných (implementovaných) zdrojov informácií, ktoré sa použijú pri rozlišovaní inštancií osôb a diel. Používateľ si môže zvoliť, ktoré zdroje budú použite.

② Ponuka výberu externých zdrojov (aktuálne okno).

③ Ponuka nastavení hodnôt pre proces rozlišovania inštancií.

④ Spustenie spracovania (transformácie) nahratých a validovaných bib. záznamov do VIVO ontológie.



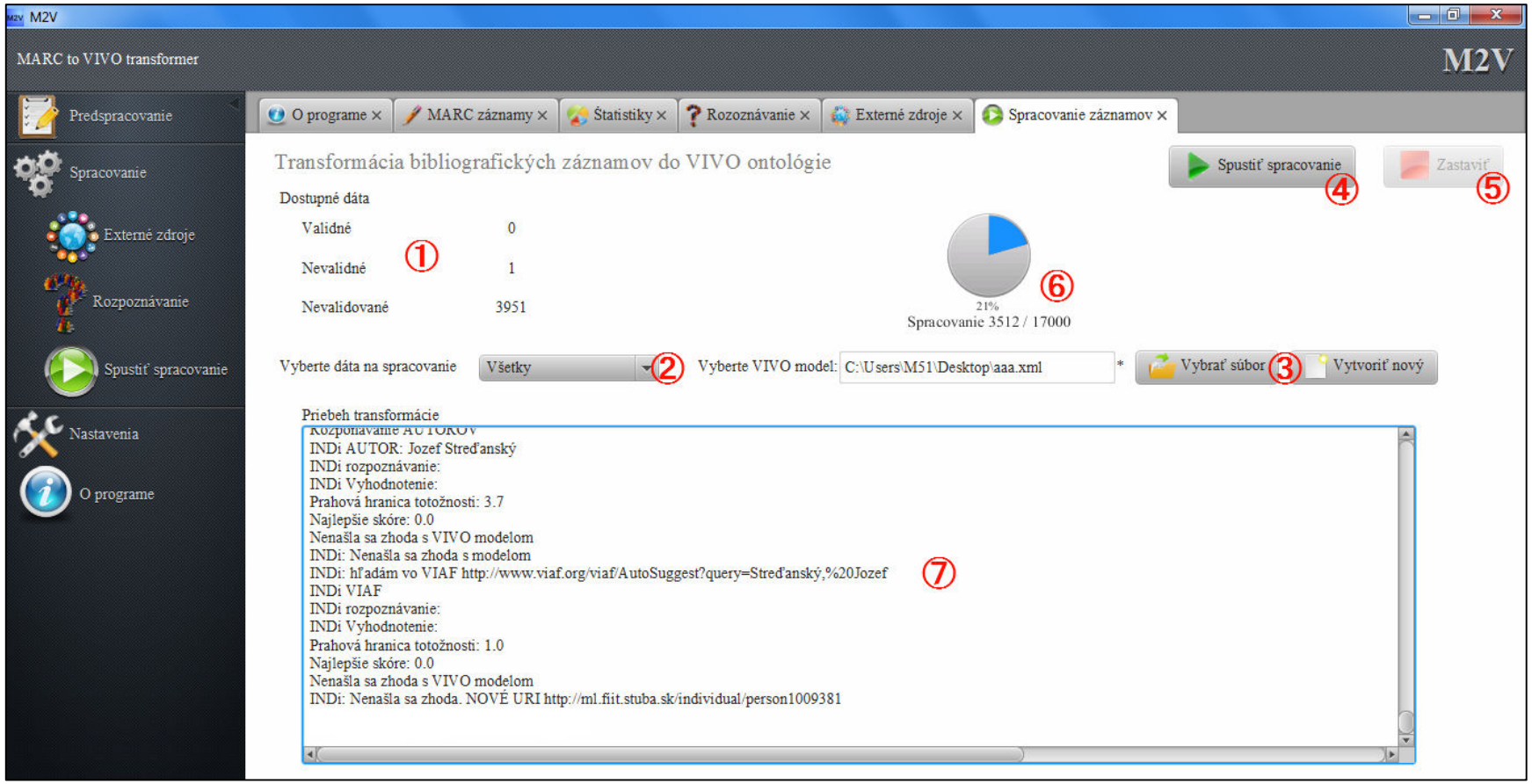
① Nastavenia hodnôt pre jednotlivé parametre porovnávania.

② Nastavenia minimálnych (prahových) hodnôt totožnosti.

③ Nastavenie hodnoty totožnosti pri porovnávaní mien osôb, názvov diel a vydavateľov. Teda vzdialenosť(podobnosť) dvoch textových reťazcov s využitím metriky Jaro-Winkler. Rozsah hodnôt je <0.0 , 1.0 >, kde napr. 0.97 znamená 97% podobnosť(zhoda).

④ Uloženie nastavení.

⑤ Načítanie pôvodných nastavení.



① Počet dostupných záznamov na spracovanie (zatriedenie do skupín: validné, nevalidné nevalidované).

② Výber typu záznamov, ktoré sa majú spracovať.

③ Výber VIVO modelu, ktorý sa použije (výber existujúceho alebo vytvorenie nového).

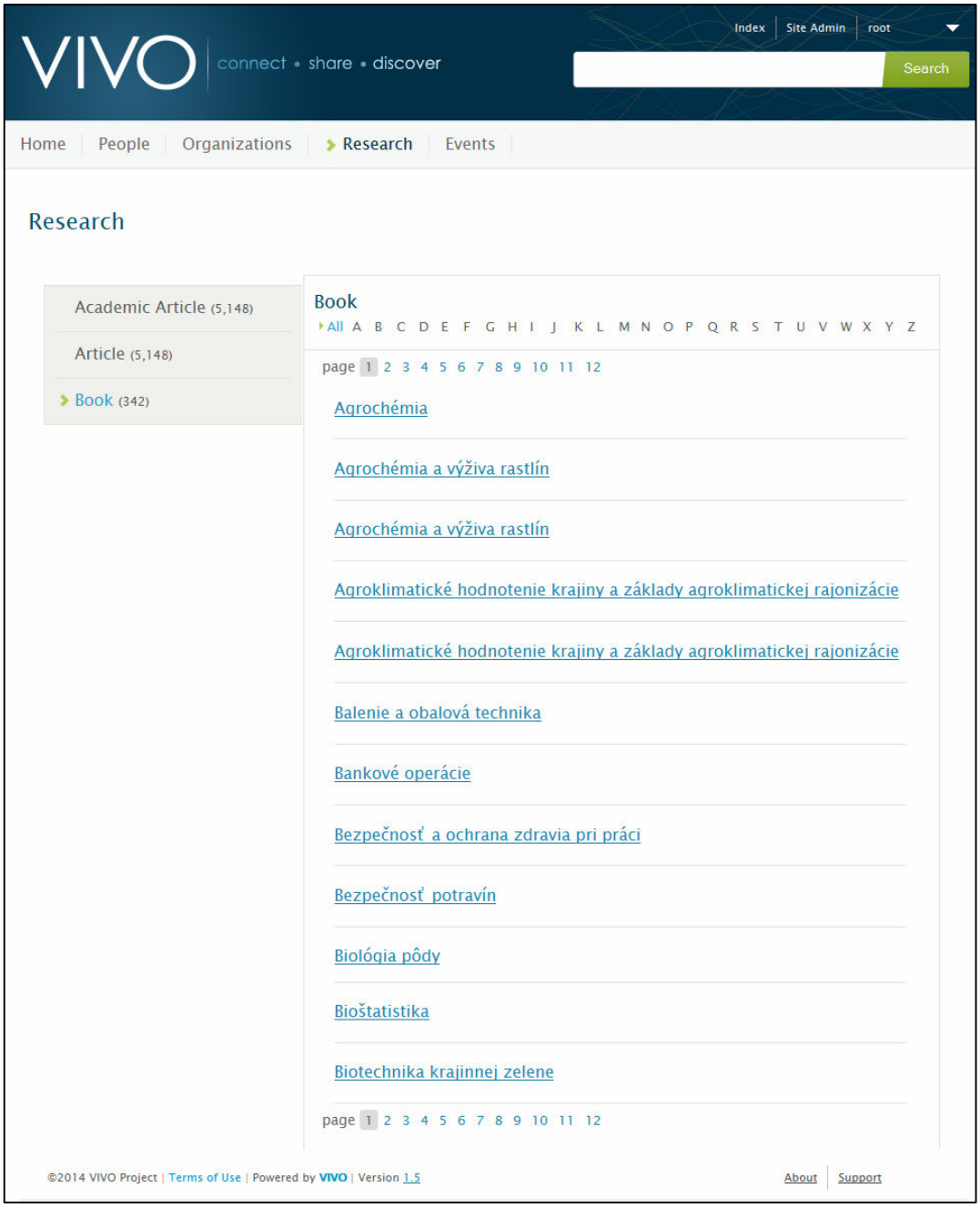
④ Spustenie spracovania vybraných bib. záznamov.

⑤ Zastavenie spracovania.

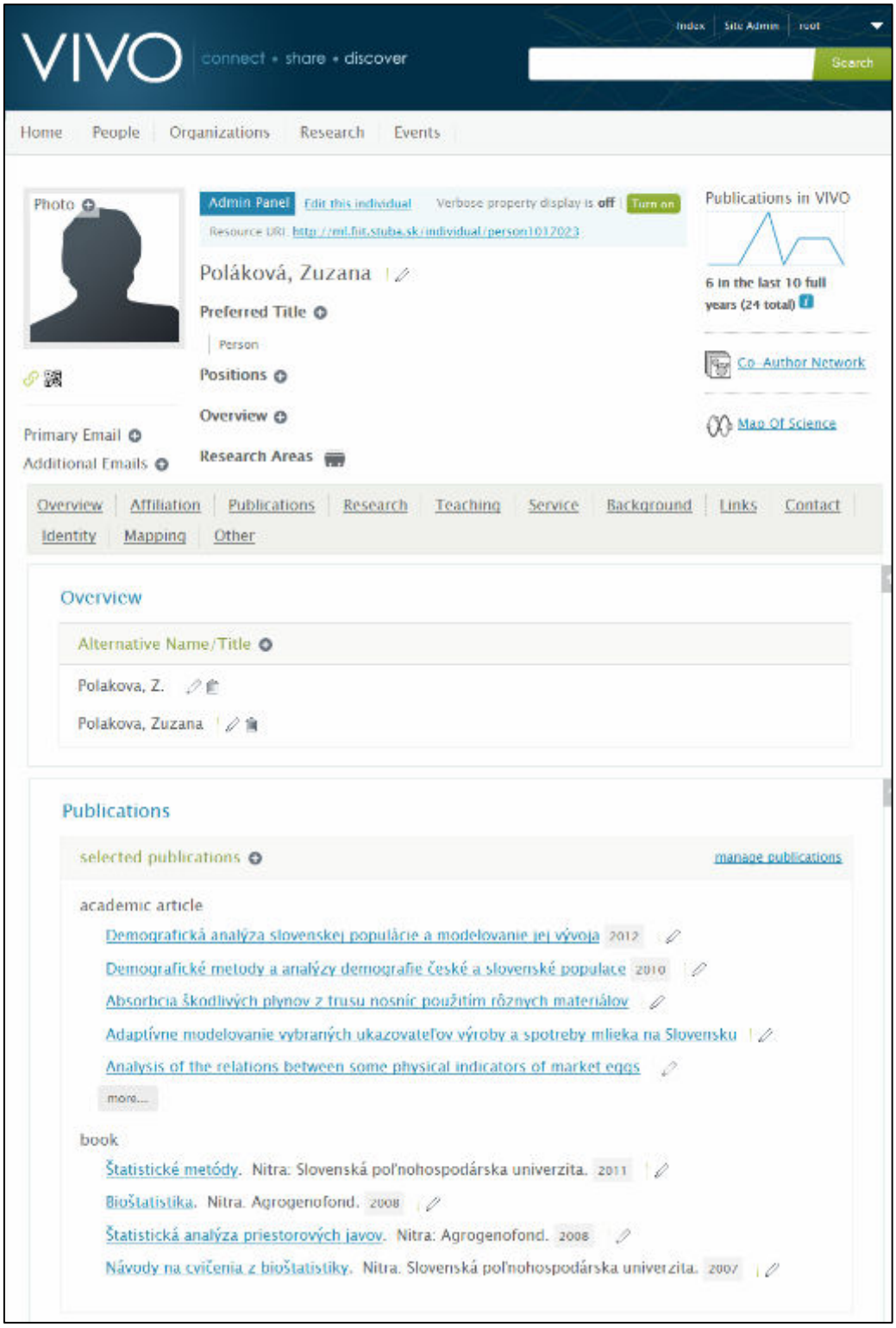
⑥ Indikátor priebehu spracovania.

⑦ Informácie o aktuálnom priebehu spracovania (log).

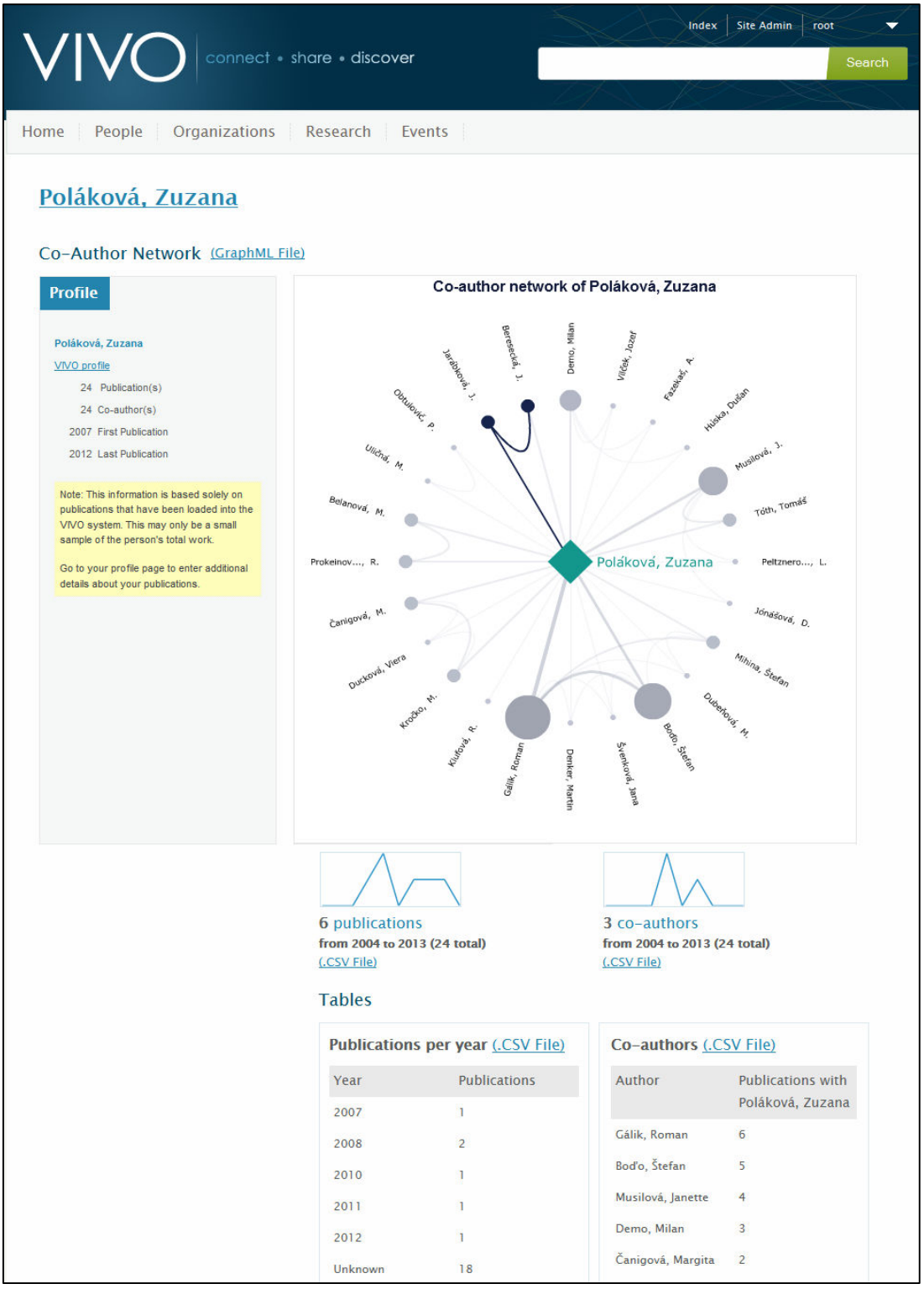
C - Ukážka VIVO rozhrania



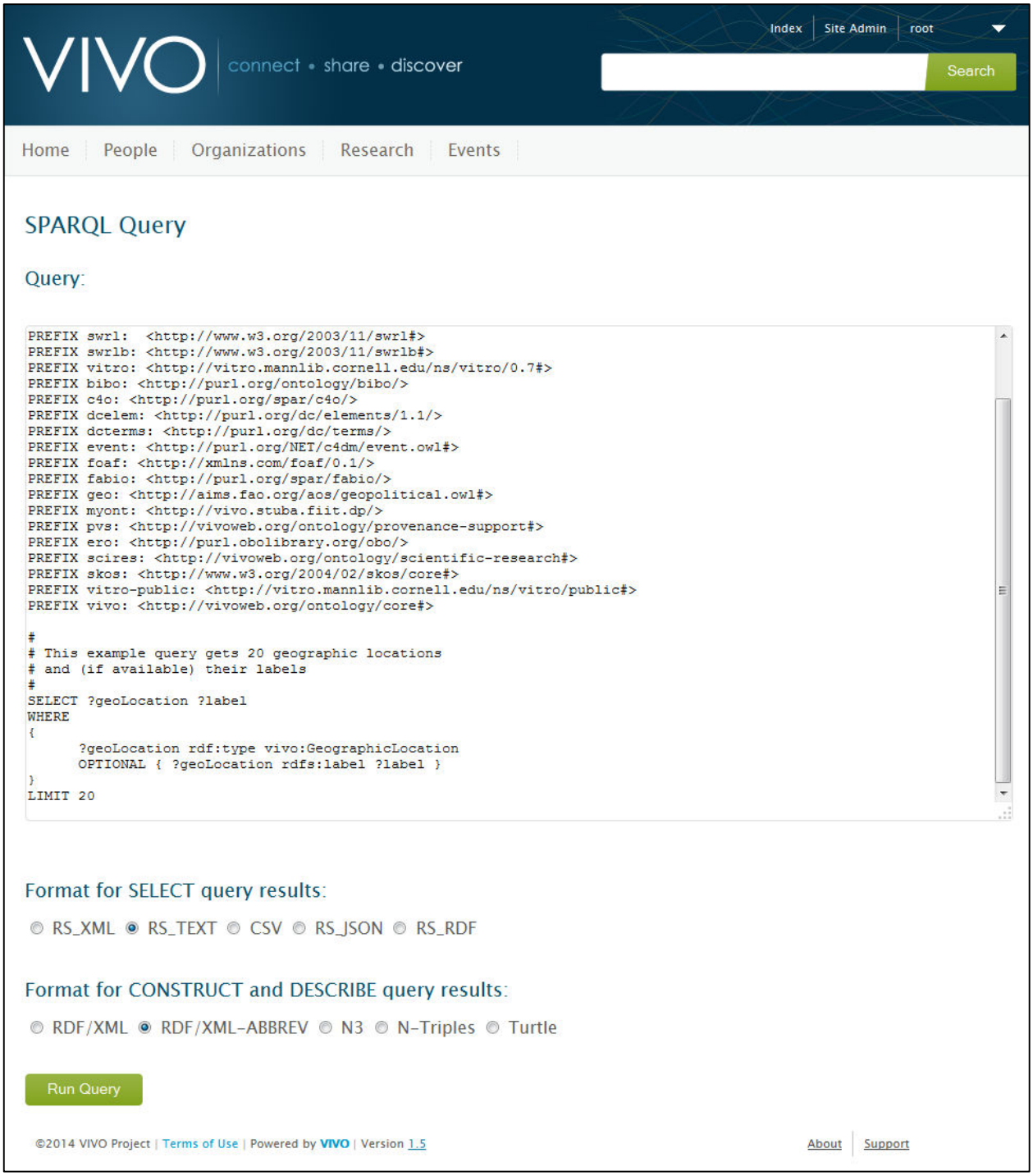
Webové rozhranie VIVO ponúka prehľadný spôsob zobrazenia importovaných dát a poskytuje možnosti na vlastnú prispôsobenie a výber prvkov a údajov, ktoré sa budú zobrazovať.



Osoby, korporácie a diela majú vo VIVO systéme vlastnú stránku, ktorá reprezentuje ich profil. Na obrázku sa nachádza profil výskumníčky obsahujúci zoznamy publikovaných diel. Okrem toho sa v profile môžu nachádzať kontaktné, identifikačné a pracovné informácie.



VIVO poskytuje nástroj na zobrazenie publikačnej činnosti osôb. Na obrázku sa nachádza graf, znázorňujúci väzby medzi skúmanou osobou a ďalšími autormi. V dolnej časti prehľadu sa nachádzajú tabuľky združujúce informácie o publikačnej činnosti v závislosti od času a zoznamy autorov, s ktorými daná osoba spoločne publikovala.



Rozhranie poskytuje editor na písanie a vykonávanie SPARQL dopytov nad dátami. Editor automaticky pred vyplní použité ontológie a základný predpis SPARQL dopytu. Výsledky dopytu môžu mať rôznu formu v závislosti od voľby používateľa.

D - Transformácia MARC záznamu do VIVO ontológie

D.1 MARC záznam

Ukážkový bib. záznam zo Slovenskej poľnohospodárskej univerzity (formát MARC 21) bol skrátený. Boli ponechané iba polia, ktoré sú transformované do VIVO ontológie.

```
<record>
<controlfield tag="001">092399</controlfield>
<controlfield tag="003">SK-NiSPK</controlfield>
<leader>00000cam--2200000-a-4500</leader>
<datafield tag="020" ind1=" " ind2=" ">
 <subfield code="a">978-80-8069-880-5 (viaz.)</subfield>
</datafield>
<datafield tag="041" ind1="0" ind2=" ">
 <subfield code="a">slo</subfield>
</datafield>
<datafield tag="245" ind1="1" ind2="0">
 <subfield code="a">Granátové jablko :</subfield>
 <subfield code="b">vedecká monografia /</subfield>
 <subfield code="c">Ján Matuškovič</subfield>
</datafield>
<datafield tag="250" ind1=" " ind2=" ">
 <subfield code="a">1. vyd.</subfield>
</datafield>
<datafield tag="260" ind1=" " ind2=" ">
 <subfield code="a">Nitra :</subfield>
 <subfield code="b">Slovenská poľnohospodárska univerzita,</subfield>
 <subfield code="c">2007</subfield>
</datafield>
<datafield tag="300" ind1=" " ind2=" ">
 <subfield code="a">128 s. :</subfield>
 <subfield code="b">grafy, obr., obr. príl., tab. ;</subfield>
</datafield>
<datafield tag="653" ind1=" " ind2=" ">
 <subfield code="9">spu_us_auth*0250164</subfield>
 <subfield code="a">ovocinárstvo</subfield>
</datafield>
<datafield tag="653" ind1=" " ind2=" ">
 <subfield code="9">spu_us_auth*0022790</subfield>
 <subfield code="a">jablká granátové</subfield>
</datafield>
<datafield tag="080" ind1=" " ind2=" ">
 <subfield code="a">634</subfield>
 <subfield code="2">2001</subfield>
</datafield>
<datafield tag="080" ind1=" " ind2=" ">
 <subfield code="a">633.879.43</subfield>
 <subfield code="2">2001</subfield>
</datafield>
<datafield tag="100" ind1="1" ind2=" ">
 <subfield code="7">spu_us_auth*0009517</subfield>
 <subfield code="a">Matušković, Ján</subfield>
 <subfield code="4">aut</subfield>
 <subfield code="u">SPUFZK06</subfield>
</datafield>
<datafield tag="040" ind1=" " ind2=" ">
 <subfield code="a">NI001</subfield>
 <subfield code="b">slo</subfield>
</datafield>
<datafield tag="970" ind1=" " ind2=" ">
 <subfield code="a">AAB</subfield>
 <subfield code="b">AMG</subfield>
</datafield>
<datafield tag="928" ind1=" " ind2=" ">
 <subfield code="7">spu_us_auth*0250013</subfield>
 <subfield code="a">Slovenská poľnohospodárska univerzita (Nitra, Slovensko)</subfield>
</datafield>
</record>
```

D.2 VIVO záznam

Ukážka transformovaného bib. záznamu z kapitoly D.1. Zelenou farbou sú vyznačené URI identifikátory generované systémom M2V v procese rozlišovania inštancií. Čiernou farbou sú vyznačené transformované alebo odvodené údaje z bib. záznamu.

<rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:j.0="http://purl.org/ontology/bibo/"
xmlns:j.1="http://xmlns.com/foaf/0.1/"
xmlns:owl="http://www.w3.org/2002/07/owl#"
xmlns:j.2="http://purl.org/dc/terms/"
xmlns:j.3="http://vivo.mydomain.edu/individual/"
xmlns:j.4="http://vivoweb.org/ontology/core#"
xmlns:myont="http://vivo.stuba.fiit.dp/"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#" >
<rdf:Description rdf:about="http://ml.fiit.stuba.sk/individual/book1019578">
<rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
<rdf:type rdf:resource="http://purl.org/ontology/bibo/Document"/>
<rdf:type rdf:resource="http://purl.org/ontology/bibo/Article"/>
<rdf:type rdf:resource="http://purl.org/ontology/bibo/AcademicArticle"/>
<rdf:type rdf:resource="http://vivoweb.org/ontology/core#InformationResource"/>
<j.2:title>**Granátové jablko**</j.2:title>
<rdfs:label>**Granátové jablko**</rdfs:label>
<myont:subtitle>**vedecká monografia**</myont:subtitle>
<j.4:freetextKeyword>**ovocinárstvo**</j.4:freetextKeyword>
<j.4:freetextKeyword>**jablká granátové**</j.4:freetextKeyword>
<j.0:isbn13 rdf:datatype="http://www.w3.org/2001/XMLSchema#string">**978-80-8069-880-5**</j.0:isbn13>
<j.4:placeOfPublication
rdf:datatype="http://www.w3.org/2001/XMLSchema#string">**Nitra**</j.4:placeOfPublication>
<j.4:dateTime>**2007**-01-01T00:00:00</j.4:dateTime>
<j.0:numPages rdf:datatype="http://www.w3.org/2001/XMLSchema#string">**>128**</j.0:numPages>
<j.4:linkURI>http://ml.fiit.stuba.sk/individual/book1019578</j.4:linkURI>
<j.0:volume rdf:datatype="http://www.w3.org/2001/XMLSchema#string">**1**</j.0:volume>
<j.4:identifier>DataSpu_EPCA***092399**</j.4:identifier>
<j.2:language rdf:datatype="http://www.w3.org/2001/XMLSchema#string">**slo**</j.2:language>
<myont:udc rdf:datatype="http://www.w3.org/2001/XMLSchema#string">**634**</myont:udc>
<myont:udc rdf:datatype="http://www.w3.org/2001/XMLSchema#string">**633.879.43**</myont:udc>
<j.4:informationResourceInAuthorship rdf:resource="http://ml.fiit.stuba.sk/relationship1040952"/>
<j.2:creator rdf:resource="http://ml.fiit.stuba.sk/individual/person1016953"/>
<j.4:publisher rdf:resource="http://ml.fiit.stuba.sk/individual/publisher1000556"/>
</rdf:Description>
<rdf:Description rdf:about="http://ml.fiit.stuba.sk/individual/person1016953">
<rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#description"/>
<rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
<rdf:type rdf:resource="http://xmlns.com/foaf/0.1/Agent"/>
<rdf:type rdf:resource="http://xmlns.com/foaf/0.1/Person"/>
<rdfs:label>**Matuškovič, Ján**</rdfs:label>
<j.1:firstName rdf:datatype="http://www.w3.org/2001/XMLSchema#string">**Ján**</j.1:firstName>
<j.1:lastName rdf:datatype="http://www.w3.org/2001/XMLSchema#string">**Matuškovič**</j.1:lastName>
<j.2:alternative rdf:datatype="http://www.w3.org/2001/XMLSchema#string">**Matuskovic, Jan**</j.2:alternative>
<j.2:alternative rdf:datatype="http://www.w3.org/2001/XMLSchema#string">**Matuškovič, J.**</j.2:alternative>
<j.2:alternative rdf:datatype="http://www.w3.org/2001/XMLSchema#string">**Matuskovic, J.**</j.2:alternative>
<j.4:identifier>**spu_us_auth*0009517**</j.4:identifier>
<j.4:linkURI>http://ml.fiit.stuba.sk/individual/person1016953</j.4:linkURI>

```
  </rdf:Description>
  <rdf:Description rdf:about="http://ml.fiit.stuba.sk/individual/publisher1000556">
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
    <rdf:type rdf:resource="http://xmlns.com/foaf/0.1/Agent"/>
    <rdf:type rdf:resource="http://xmlns.com/foaf/0.1/Organization"/>
    <rdf:type rdf:resource="http://vivoweb.org/ontology/core#Publisher"/>
    <j.4:linkURI>http://ml.fiit.stuba.sk/individual/publisher1000556</j.4:linkURI>
    <rdfs:label>Slovenská pol'nohospodárska univerzita</rdfs:label>
    <j.4:identifier>spu_us_auth*0250013</j.4:identifier>
    <j.4:publisherOf rdf:resource="http://ml.fiit.stuba.sk/individual/book1019578"/>
  </rdf:Description>
  <rdf:Description rdf:about="http://ml.fiit.stuba.sk/relationship1040952">
    <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#description"/>
    <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#Thing"/>
    <rdf:type rdf:resource="http://vivoweb.org/ontology/core#Relationship"/>
    <rdf:type rdf:resource="http://vivoweb.org/ontology/core#Authorship"/>
    <j.4:linkedAuthor rdf:resource="http://ml.fiit.stuba.sk/individual/person1016953"/>
    <j.4:linkedInformationResource rdf:resource="http://ml.fiit.stuba.sk/individual/book1019578"/>
    <j.4:linkURI>http://ml.fiit.stuba.sk/relationship1040952</j.4:linkURI>
  </rdf:Description>
</rdf:RDF>
```

E - Použité ontológie

#	Názov ontológie	Menný priestor	Prefix
1	Bibontology	http://purl.org/ontology/bibo/	bibo
2	Citation Counting and Context Characterization Ontology	http://purl.org/spar/c4o/	c4o
3	Dublin Core elements	http://purl.org/dc/elements/1.1/	dcelem
4	Dublin Core terms	http://purl.org/dc/terms/	dcterms
5	Event Ontology	http://purl.org/NET/c4dm/event.owl#	event
6	FOAF	http://xmlns.com/foaf/0.1/	foaf
7	FRBR-aligned Bibliographic Ontology	http://purl.org/spar/fabio/	fabio
8	geopolitical.owl	http://aims.fao.org/aos/geopolitical.owl#	geo
9	provenance support	http://vivoweb.org/ontology/provenance-support#	pvs
10	Research Resources (eagle-i)	http://purl.obolibrary.org/obo/	ero
11	Scientific Research	http://vivoweb.org/ontology/scientific-research#	scires
12	SKOS (Simple Knowledge Organization System)	http://www.w3.org/2004/02/skos/core#	skos
13	Vitro Public Ontology	http://vitro.mannlib.cornell.edu/ns/vitro/public#	vitro-public
14	VIVO core	http://vivoweb.org/ontology/core#	vivo

F - Prevodová tabuľka
F.1 MARC 21

Tabuľka č. 12 Prevod polí a podpolí z formátu MARC 21 do VIVO ontológie.

Pole	Podpole	RDF	Poznámka
001		core:identifier	Evidenčné číslo záznamu
008		vivo:dateTime	7-10 pozícia
020	a	bibo:isbn10 bibo:isbn13	ISBN10 ISBN13
022	a	bibo:issn	ISSN
041	a	dcterams:language	Kód jazyka textu diela
	b	dcterams:language	Kód jazyka zhrnutia alebo abstraktu
080	a	myont:udc	Číslo medzinárodného desatinného triedenia
100	a	foaf:firstname foaf:lastname rdfs:label	Osobné meno
	d	myont:yearOfBirth myont:deathYear	dátum narodenia dátum úmrtia
	4	dcterms:creator vivo:editor bibo:translator	Kód roly
	7	core:identifier	Identifikátor v rámci knižničného systému
242	a	dcterms:alternative	Preklad názvu - paralelný názov
245	a	dcterms:title vivo:title rdfs:label	Názov diela
	b	myont:subtile	Zvyšok názvu
246	a	dcterms:alternative	Iná forma názvu (paralelný názov)
250	4	bibo:volume	Poradové číslo vydania
260	a	vivo:placeOfPublication	Miesto publikovania, distribúcie
	b	rdfs:label	Názov vydavateľa, distribútora
	c	vivo:dateTime	Dátum publikovania, distribúcie
300	a	bibo:numPages	Fyzický popis - počet strán. Strany je potrebné extrahovať v závislosti od spôsobu zápisu informácií v danom poli.
500	a	bibo:abstract	Abstrakt
600	a	vivo:freetextKeyword	Tematické heslo - Osobné meno
610	a	vivo:freetextKeyword	Tematické heslo - Korporatívne meno
630	a	vivo:freetextKeyword	Tematické heslo - Meno zhromaždenia
650	a	vivo:freetextKeyword	Tematické heslo - Unifikovaný názov
651	a	vivo:freetextKeyword	Tematické heslo - Geografický názov
700	a	foaf:firstname foaf:lastname rdfs:label	Osobné meno
	d	myont:yearOfBirth myont:deathYear	dátumy narodenia, úmrtia
	4	dcterms:creator	Kód roly

		vivo:editor bibo:translator	
	7	core:identifier	Identifikátor v rámci knižničného systému
765	t	dcterms:alternative	Originálny názov (v prípade prekladu)
856	u	dcterms:source	Elektronická lokalizácia a prístup
928	7	core:identifier	Identifikátor vydavateľa v knižničnom systéme
970	a	rdf:type	Kód typu publikácie (kniha, článok) Na základe tejto hodnoty sa rozlišuje podtrieda diela: bibo:Document bibo:AcademicArticle

F.2 UNIMARC

Tabuľka č. 13 Prevod polí a podpolí z formátu UNIMARC do VIVO ontológie.

Pole	Podpole	RDF	Poznámka
001		core:identifier	Evidenčné číslo záznamu
010	a	bibo:isbn10 bibo:isbn13	ISBN10 ISBN13
011	a	bibo:issn	ISSN
101	a	dcterams:language	Kód jazyka textu diela
200	a	dcterms:title vivo:title rdfs:label	Názov diela
	d	dcterms:alternative	Paralelný názov
	b	myont:subtile	Zvyšok názvu
205	a	bibo:volume	Poradové číslo vydania
210	a	vivo:placeOfPublication	Miesto publikovania, distribúcie
	c	rdfs:label	Názov vydavateľa, distribútora
	d	vivo:dateTime	Dátum publikovania, distribúcie
215	a	bibo:numPages	Fyzický popis - počet strán. Strany je potrebné extrahovať v závislosti od spôsobu zápisu informácií v danom poli.
541	a	dcterms:alternative	Originálny názov (v prípade prekladu)
600	a	vivo:freetextKeyword	Tematické heslo - Osobné meno
601	a	vivo:freetextKeyword	Tematické heslo - Korporatívne meno
605	a	vivo:freetextKeyword	Tematické heslo - Názov diela
607	a	vivo:freetextKeyword	Tematické heslo - Geografický názov
610	a	vivo:freetextKeyword	Tematické heslo - Nekontrolovaná hodnota
615	a	vivo:freetextKeyword	Tematické heslo - Kategória
675	a	myont:udc	Číslo medzinárodného desatinného triedenia
700	b	foaf:firstname	Osobné meno
	a	foaf:lastname	Priezvisko
	f	myont:yearOfBirth myont:deathYear	dátum narodenia dátum úmrtia
	4	dcterms:creator vivo:editor bibo:translator	Kód roly
	3	core:identifier	Identifikátor v rámci knižničného systému

701	b	foaf:firstname	Osobné meno
	a	foaf:lastname	Priezvisko
	f	myont:yearOfBirth myont:deathYear	dátum narodenia dátum úmrtia
	4	dcterms:creator vivo:editor bibo:translator	Kód roly
	3	core:identifier	Identifikátor v rámci knižničného systému
702	b	foaf:firstname	Osobné meno
	a	foaf:lastname	Priezvisko
	f	myont:yearOfBirth myont:deathYear	dátum narodenia dátum úmrtia
	4	dcterms:creator vivo:editor bibo:translator	Kód roly
	3	core:identifier	Identifikátor v rámci knižničného systému
856	u	dcterms:source	Elektronická lokalizácia a prístup
928	3	core:identifier	Identifikátor vydavateľa v knižničnom systéme
970	a	rdf:type	Kód typu publikácie (kniha, článok) Na základe tejto hodnoty sa rozlišuje podtrieda diela: bibo:Document bibo:AcademicArticle

F.3 VIVO Triedy

Triedy dielo, osoba, vydavateľ musia nadobudnúť definovaný typ (rdf:type) v rámci VIVO ontológie aby ich VIVO systém vedel správne identifikovať, spravovať a zobrazovať. Uvedené typy sú uvedené v tabuľke č. 14.

Tabuľka č. 14 Definované triedy a ich vlastnosti v ontológií VIVO.

Trieda	MARC 21		UNIMARC		rdf:type
	Pole	podpole	pole	podpole	
Osoba	100 700	a	700 701 702	b,a	rdf:description owl:Thing foaf:Agent foaf:Person
Vydavateľ (organizácia)	260	b	210	c	owl:Thing foaf:Agent foaf:Organization vivo:Publisher
Kniha (dielo)	245	a	200	a	owl:Thing bibo:Document bibo:book vivo:InformationResource
Akademický článok (dielo)	245	a	200	4	owl:Thing bibo:Document bibo:Article bibo:AcademicArticle vivo:InformationResource

G - Inštalačná príručka

G.1 Inštalácia systému M2V

Systém M2V je distribuovaný ako spustiteľný program. Systém sa inštaluje zo súboru M2V_install.exe, ktorý sa nachádza na priloženom elektronickom médiu. Inštalácia je priamočiara a nevyžaduje žiadne používateľské vstupy.

Po inštalácií programu, pred jeho prvým spustením je potrebné vytvoriť databázu M2VDB na MySQL serveri.

Skript na vytvorenie databázy sa nachádza sa nachádza na priloženom elektronickom médiu:
/Installation/DB/M2V_DB_empty.sql.

Používateľ môže zvoliť aj druhý priložený skript:
/Installation/DB/M2V_DB_data.sql, ktorý obsahuje okrem schémy databázy aj určité množstvo testovacích dát.

Po spustení programu je potrebné vyplniť prihlasovacie údaje na vytvorenie spojenia s databázou:
Bočné menu: Nastavenia => Databáza.

G.2 Inštalácia systému VIVO

Podrobný návod na inštaláciu systému VIVO sa nachádza na priloženom elektronickom médiu v priečinku: /Installation/VIVO. Súbor sa nazýva VIVO_Release_V1.5_Installation_Guide.pdf.

Pred inštaláciou systému VIVO je potrebné nainštalovať nasledovný softvér:

- Java (SE) 1.6.x http://java.sun.com
- Apache Tomcat 6.x alebo 7.x http://tomcat.apache.org
- Apache Ant 1.8 alebo vyšší, http://ant.apache.org
- MySQL 5.1 alebo vyšší, http://www.mysql.com

Následne je potrebné nastaviť premenné prostredia `JAVA_HOME` a `ANT_HOME`.
Zvyšné pokyny na inštaláciu VIVO systému sú uvedené v spomínanom návode.

G.3 Import dát do VIVO systému

Import dát sa uskutočňuje po prihlásení do VIVO systému v časti:

Site Admin → Add/Remove RDF data.

Následne je potrebné zvoliť možnosť:

- add instance data (supports large data files)

a vybrať príslušný korpus (VIVO model), ktorý má byť importovaný do systému.

Používateľ musí byť prihlásený ako administrátor systému. Import môže trvať niekoľko desiatok minút.

H - Obsah elektronického média

/README.TXT	obsah elektronického média
/Installation/M2V/dist/M2V.exe	spustiteľný súbor systému
/Installation/VIVO/ vivo-rel-1.5.zip	archív obsahujúci inštalačné súbory systému VIVO
/Installation/VIVO/VIVO_Release_V1.5_Installation_Guide.pdf	inštalačná príručka sys. VIVO
/Installation/DB/M2V_DB_data.sql	korpus databázy M2VDB aj dátami (bib. záznamov SPU)
/Installation/DB/M2V_DB_empty.sql	prázdna databáza
/Project/M2V.zip	java projekt, export z Eclipse IDE
/Data/SPU_data.xml	bib. záznamov (MARC/XML)
/Data/VIVO_model_SPU.xml	bib. záznamy z SPU transformované do VIVO ontológie
/Diploma Thesis/Diplomova_Praca_Marek_Loderer.docx	dokument vo formáte .docx
/Diploma Thesis/Diplomova_Praca_Marek_Loderer.pdf	dokument vo formáte .pdf