ZÁPADOČESKÁ UNIVERZITA V PLZNI
FAKULTA APLIKOVANÝCH VĚD
KATEDRA MATEMATIKY

Porovnání projektů Wikidata a DBpedia jako zdrojů prostorových dat

Bakalářská práce

Comparison of Wikidata and DBpedia projects as spatial data sources

Bachelor thesis

Jan Macura <jmacura@students.zcu.cz>

Vedoucí práce
Ing. Mgr. Otakar Čerba, Ph.D.

Plzeň, 2016

Prohlášení

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

V Plzni dne ....

Jan Macura

Poděkování

Rád bych poděkoval zejména Ing. Mgr. Otakaru Čerbovi, Ph.D. za cenné rady a připomínky a za trpělivost v průběhu vedení této práce. Taktéž děkuji Jiřímu Sedláčkovi, BA (Hons), Markusi Krötzschi, M.Sc. a Stasu Malyshevovi za poskytnuté rady a informace, týkající se projektu Wikidata. Zároveň bych rád poděkoval svým rodičům za soustavnou podporu při tvorbě této práce i v celém průběhu studia a mé partnerce za důvěru.

ZÁPADOČESKÁ UNIVERZITA V PLZNI
Fakulta aplikovaných věd
Akademický rok: **2015/2016**

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení:	**Jan MACURA**
Osobní číslo:	**A12B0304P**
Studijní program:	**B3602 Geomatika**
Studijní obor:	**Geomatika**
Název tématu:	**Porovnání projektů Wikidata a DBpedia jako zdrojů prostorových dat**
Zadávající katedra:	**Katedra matematiky**

Zásady pro vypracování:

1. Co jsou to Wikidata a co je DBpedia. Vznik. Vývoj.
2. Struktura Wikidat, jejich API, údržba a přesnost dat.
3. Struktura DBpedie, přístup k ní a aktuálnost dat.
4. Možnosti vizualizace dat.
5. Konkrétní příklady aplikace.

Rozsah grafických prací: **dle potřeby**

Rozsah kvalifikační práce: **cca 20 stran**

Forma zpracování bakalářské práce: **tištěná**

Seznam odborné literatury:

- **ISMAYLOV, Ali, KONTOKOSTAS, Dimitris, AUER, Sören, LEHMANN, Jens, HELLMANN, Sebastian. Wikidata through the Eyes of DBpedia. In: ISWC 2015 Proceedings. Bethlehem: CoRR, 2015, abs/1507.04180.**
- **VRANDEČIĆ, Denny; KRÖTZSCH, Markus. Wikidata: A Free Collaborative Knowledgebase. Communications of the ACM. 2014, Vol. 57 No. 10, 7885. DOI 10.1145/2629489.**
- **AUER, Sören; BIZER, Christian; KOBILAROV, Georgi; LEHMANN, Jens; CYGANIAK, Richard; IVES, Zachary. DBpedia: A Nucleus for a Web of Open Data. The Semantic Web. 2007. Volume 4825, 722735. ISSN 0302-9743. ISBN 978-3-540-76298-0.**

Vedoucí bakalářské práce: **Ing. Mgr. Otakar Čerba, Ph.D.**
Katedra matematiky

Datum zadání bakalářské práce: **1. října 2014**

Termín odevzdání bakalářské práce: **31. května 2016**

Doc. RNDr. Miroslav Lávička, Ph.D.
děkan

L.S.

Prof. RNDr. Pavel Drábek, DrSc.
vedoucí katedry

V Plzni dne 1. října 2015

Abstrakt

Tato práce se zaměřuje na problematiku Linked Open Data projektů Wikidata a DBpedia, se zaměřením na prostorová data. Cílem práce je nalezení způsobů, jak je možné prostorových dat z těchto projektů využít a jak je vizualizovat. Obsahem práce je podrobné srovnání struktury, přístupnosti, způsobu popisu prostorových jevů a obsáhlosti obou databázových projektů. Dále práce rozebírá specifika, která se s prostorovými daty v řešených databázích pojí, a nabízí ukázku, jak lze data využít v kartografické aplikaci. V ukázkové aplikaci jsou použity dva dotazy v jazyce SPARQL, které umožní najít a v mapovém okně zobrazit *a*) města, v jejichž čele stojí žena a *b*) místa, ve kterých se narodili nositelé Nobelovy ceny. Zdrojový kód této aplikace je otevřený, tudíž kdokoliv se může inspirovat a aplikaci adaptovat pro své účely.

Klíčová slova

DBpedia, Wikidata, Sémantický web, otevřená data, linked data, linked open data, ontologie, databáze, porovnání, RDF, OWL, prostorová data, big data, vizualizace

Abstract

This thesis focuses on the issue of Linked Open Data projects Wikidata and DBpedia, with the main aim on spatial data. The objective of this work is to find a way, how it is possible to use spatial data from these projects and how to visualise them. The content of this work is detailed comparison of structure, accessibility, description of spatial features and extensiveness of both database projects. The thesis further explains the specificities connected with spatial data in those databases and offers an example, how to use the data in a cartographic application. In the example application are present two queries in the SPARQL language, which one allows to find and display *a*) cities with female mayor and *b*) birthplaces of Nobel prize laureates, in the map window. The code of this application is open source, hence anyone can get inspired and adapt the application for one's purpose.

Keywords

DBpedia, Wikidata, Semantic web, open data, linked data, linked open data, ontology, database, comparison, RDF, OWL, spatial data, big data, visualisation

Obsah

Seznam obrázků

Seznam tabulek

Seznam ukázek

Seznam použitých zkratek

- AJAX – Asynchronous JavaScript and XML
- angl. – anglicky
- API – Application Programming Interface
- BLOB – Binary Large Object
- CC – Creative Commons
- CC0 – Creative Commons Public Domain Dedication
- CC-BY-SA – Creative Commons Attribution-ShareAlike License
- CORS – Cross-origin Resource Sharing
- CSS – Cascading Style Sheets
- CSV – Comma-separated Values
- DCMI – Dublin Core Metadata Initiative
- DE-9IM – Dimensionally Extended nine-Intersection Model
- DOM – Document Object Model
- FOAF – Friend of a Friend
- GeoJSON – JavaScript Object Notation for geographic objects
- GeoSPARQL – A Geographic Query Language for RDF Data
- GFDL – GNU Free Documentation License
- GNU – GNU's Not Unix!
- GPL – General Public License
- HTML – HyperText Markup Language
- HTTP – Hypertext Transfer Protocol
- IRI – Internationalized Resource Identifier
- JSON – JavaScript Object Notation
- JSONP – JSON with padding
- LOD – Linked Open Data
- MPLv2 – Mozilla Public License Version 2.0
- OS – operační systém
- OWL – Web Ontology Language
- OWL/XML – Web Ontology Language in Extensible Markup Language
- PHP – PHP: Hypertext Preprocessor
- RAM – Random Access Memory
- RDF – Resource Description Framework
- RDFa – Resource Description Framework in attributes

- RDFS – RDF Schema
- RDF/XML – Resource Description Framework in Extensible Markup Language
- REST – Representational State Transfer
- SKOS – Simple Knowledge Organization System
- SPARQL – SPARQL Protocol And RDF Query Language
- SQL – Structured Query Language
- TIN – Triangulated irregular network
- URI – Uniform Resource Identifier
- URL – Uniform Resource Locator
- WGS84 – World Geodetic System 1984
- WKT – Well-known text
- WMF – Wikimedia Foundation
- WWW – World Wide Web
- XML – Extensible Markup Language
- YAGO – Yet Another Great Ontology

Úvod

Tato práce se věnuje klíčovým projektům, realizujícím koncept Linked Open Data – znalostní bázi DBpedia[1] a databázi Wikidata[2]. První jmenovaný projekt, DBpedia, zpracovává obsah internetové encyklopedie Wikipedia[3] a převádí jej do strojově čitelné podoby Linked Data. DBpedia zároveň obsahuje odkazy do dalších obdobných databází, kterých stále přibývá, stejně jako zpětného odkazování z jiných zdrojů, a DBpedia se tak stala jádrem Sémantického webu. Druhý projekt, Wikidata, má za úkol uchovávat strojově čitelná data odděleně od souvislého textu, který je obsahem encyklopedie Wikipedia, a umožnit tak jejich jednoduchou přístupnost, bez nutnosti pozdější extrakce.

Cílem této práce je najít využití potenciálu, který obě tyto databáze otevřených dat nabízejí pro prostorová data. DBpedia odkazuje do ontologie Geonames[4], obsahuje zeměpisné souřadnice extrahované z encyklopedie Wikipedia a více než 1 000 000 dalších. Taktéž velké množství položek v databázi Wikidata je doplněno o zeměpisné souřadnice. Tato práce se snaží shrnout, jaké jsou současné možnosti kartografické vizualizace prvků těchto databází a jaká pro ni existují omezení. Hlavním cílem práce je vzájemné porovnání obou projektů z pohledu využitelnosti prostorových dat v nich uložených. Vzhledem k neúplné dokumentaci obou projektů je též vedlejším účelem této práce přehledně popsat alespoň část jejich struktury a obsahu, a zvýšit tak přístupnost obou datových bází. V neposlední řadě je motivací k vytvoření této práce poskytnutí dostatečného množství informací o těchto projektech i v českém jazyce.

Tato práce je členěna následovně: První kapitola obsahuje úvod do problematiky Linked Data a stručný popis struktury obou projektů. Ve druhé kapitole jsou diskutovány existující metody porovnávání znalostních bází, a na jejich základě je navržena použitá metoda porovnání projektů Wikidata a DBpedia. Ve třetí kapitole je věnován prostor samotnému porovnání vybraných parametrů obou databází. Čtvrtá kapitola je zaměřena na kartografickou vizualizaci dat, její omezení a na existující aplikace, které těchto dat využívají. V páté kapitole je učiněno shrnutí výsledků práce. K textu práce jsou přiloženy použité dotazy v jazyce SPARQL, doprovodné ilustrace ve větším měřítku pro lepší čitelnost a zdrojové kódy vytvořených programů.

[1] http://wiki.dbpedia.org/
[2] https://www.wikidata.org/wiki/Wikidata:Main_Page
[3] https://en.wikipedia.org/wiki/Main_Page
[4] http://www.geonames.org/

1 Popis Linked Open Data projektů

1.1 Koncept Linked Open Data

Již od svého vzniku v roce 1989 je World Wide Web (zkr. WWW, Web) živý a vyvíjející se systém. Během deseti let se struktura a funkčnost WWW změnily natolik, že se tento „nový Web“ začal označovat jako Web 2.0. Zásadní byl obrat, kdy se ze statických stránek, ležících v síti a čekajících, na své přečtení, staly dynamické stránky, reagující na podněty uživatele. Postupem času se pojem Web 2.0 začal spojovat se stavem, kdy uživatel je zároveň tvůrcem obsahu, a tuto fázi dobře ilustrují nejprve četná internetová diskusní fóra a později sociální sítě.

Na internetu se začalo hromadit nepřeberné množství obsahu. Například na anglické verzi internetové encyklopedie Wikipedia proběhne přibližně 100 000 editací za den[1]. V takové záplavě informací je téměř nemožné se zorientovat a hledat v ní odpovědi na konkrétní otázky je velmi obtížné. Revolučním se v tomto směru stal koncept Linked Data, který pro data stanovuje základní pravidla: Každá položka zveřejňovaných dat musí být jednoznačně identifikovatelná; při jejím prohlížení skrze Web musí uživatel dostat smysluplné informace ve standardizované podobě; a pokud informace v položce zmiňují jinou položku dat, musí na ni odkazovat pomocí jejího jednoznačného identifikátoru. Tak vznikne propojená síť dat, která lze logicky procházet a hledat mezi nimi vztahy a vazby. Pokud pro data navíc platí, že jsou dostupná veřejně a za podmínek svobodné licence, hovoří se o tzv. Linked Open Data.

Linked Data zásadně ovlivnila vznik tzv. Sémantického webu, což označuje Web, kde nejsou dokumenty čitelné pouze pro člověka, ale díky doplnění jejich jednotlivých částí o sémantické informace a vzájemnému propojení, jsou přístupné také pro automatizované procházení. Sémantický web je často chápán jako zásadní pro rozvoj Webu 2.0, někdy však bývá označován přímo jako Web 3.0.

Tim Berners-Lee rozdělil informace dostupné prostřednictvím internetu do pěti tříd podle způsobu jejich zpřístupnění, přičemž sledoval zejména výše zmíněné vlastnosti, které definují Linked Data a Linked Open Data. Takto vzniklá stupnice se nazývá též *5hvězdičková stupnice*, podle přidělovaných hvězd za každou splněnou podmínku přístupnosti. [8]

Jednou hvězdou jsou označené dokumenty dostupné veřejně. To zahrnuje obrázky nebo dokumenty ve formátu PDF, které jsou čitelné pouze člověkem. Takovými dokumenty jsou například články odkazované v části Reference této práce, nebo tato práce samotná, tak jak je archivována v univerzitní knihovně.

[1] Podle https://stats.wikimedia.org/EN/SummaryEN.htm cca 3,5 mil. editací za měsíc.

Dvě hvězdy označují dokument, který je krom veřejné přístupnosti také strojově čitelný. To může znamenat například tabulky, prvky Webu doplněné o sémantické informace pomocí formátů RDF-a, Microdata, atp. nebo dokumenty založené na XML.

Třemi hvězdami jsou označeny dokumenty, které jsou kromě předchozích dvou vlastností dostupné v otevřeném, standardizovaném a dokumentovaném formátu. Tuto vlastnost splňují například dokumenty ve formátu DocBook, OpenDocument, CSV nebo tato práce ve formátu LaTeX.

Dokumenty, splňující čtyři hvězdy, jsou někdy chybně označovány jako Linked Data. Odpovídajícím termínem by pro ně bylo „Linkable Data“, ten se však téměř nepoužívá. Spíše než o dokumentech je též vhodnější mluvit o datech, neboť pro splnění hodnocení 4 hvězd musí být každá položka datového souboru navíc identifikovatelná svým URI, tedy je možno na ni jednoznačně odkázat, a pomocí tohoto URI by měla být dereferencovatelná. Pojem dereference obecně znamená následování reference k nalezení objektu. V tomto kontextu je dereferencovatelná položka taková, která při přístupu klienta na jí přiřazené URI vrátí svůj obsah ve strukturované formě. K popisu takto zpřístupněných dat se nejčastěji využívá formátu RDF, který bude podrobněji vysvětlen později, může ale jít stále i o patřičně strukturovanou tabulku ve formátu OpenDocument.

Poslední, pátou, hvězdou lze označit datovou sadu, kterou kromě předchozích vlastností doplňuje propojení vlastních dat s jinými datovými sadami. Toto vzájemné propojení vede ke vzniku komplexní datové sítě, někdy označované jako LOD cloud. Vzájemné odkazování mezi databázemi navíc předchází zbytečné duplikaci stejných tvrzení na více místech. Všech těchto pět podmínek splňují například databáze DBpedia a Wikidata.

1.2 Vývoj encyklopdie Wikipedia a vznik projektů DBpedia a Wikidata

Když 15. ledna 2001 Jimmy Wales a Larry Sanger zakládali internetovou encyklopedii *Wikipedia* (později česky *Wikipedie*), šlo jen o doplňkový projekt starší encyklopedie *Nupedia*, která nabízela svůj obsah jako svobodný. Zatímco *Nupedia*, která byla vytvářena v sedmistupňovém editačním procesu, převážně vybranými odborníky, přestala být postupně aktualizována, až byla v roce 2003 opuštěna úplně, *Wikipedie*, která byla vystavěna na principu, umožňujícím editaci jakémukoliv uživateli se dočkala velkého úspěchu a v roce 2014 patřila mezi 10 nejpopulárnějších serverů vůbec. [1, 2]

Wikipedie má 287 jazykových mutací, přičemž 125 z nich obsahuje více jak 10 000 článků. Její výjimečnost a důležitost pro geografii, a prostorová data vůbec, dokládá

například i testovací analýza, na jejíž základě byl prozatím nejexaktněji prokázán Toblerův první zákon geografie (podrobněji v [9]). Vedle *Wikipedie* postupně vzniklo dalších 9 sesterských projektů, které se zaměřují například na sběr svobodných multimédií, knih, citátů nebo zpráv. Společně jejich organizaci a strukturu řídí nezávislá Nadace Wikimedia (anglicky Wikimedia Foundation, WMF).

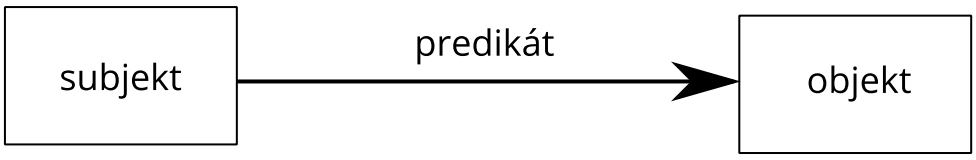
V projektech WMF je uchováno neobvyklé množství informací a lidských vědomostí. Naneštěstí, tyto informace nejsou uchovávány v žádné strukturované podobě, použitelné pro automatické zpracování a snadnou strojovou přístupnost. To se pokusili změnit v roce 2007 vědci z univerzit v Lipsku a Mannheimu, když spustili projekt s názvem DBpedia. DBpedia extrahuje maximální množství dat z Wikipedie do strukturované databáze, která je přístupná v režimu open data.

Stejného nedostatku si začala být vědoma i WMF, a tak v roce 2012 spustila zatím nejnovější sesterský projekt s názvem Wikidata. Wikidata jsou rozsáhlou webovou znalostní bází, která má sloužit jako podklad pro ostatní projekty a umožnit jednoduchý přístup ke strojově čitelným datům.

1.3 Struktura databáze DBpedia

1.3.1 Datový model databáze DBpedia

DBpedia uchovává data v abstraktním datovém modelu, založeném na formátu RDF. Datový model má podobu orientovaného, hranově i vrcholově ohodnoceného grafu. RDF trojice (též triplet) subjekt (podmět) – predikát (přísudek) – objekt (předmět), tak má z pohledu teorie grafů strukturu dvou vrcholů, spojených orientovanou hranou dané hodnoty (vlastnosti). Subjekt a predikát jsou vždy definovány svou URI. Objektem může být další RDF trojice, URI, prostý text, nebo jiný definovaný datový typ. [3] Tento základní vztah je zobrazen na obrázku 1.



Obrázek 1: Resource Description Framework trojice.

Databáze DBpedia vzniká extrakcí strukturovaných dat z mnohojazyčné encyklopedie Wikipedia. Nové verze vycházejí průměrně jednou, až dvakrát ročně. Od původní jednotné databáze se v roce 2011 přešlo k systému lokálních databází, rozdělených

podle jazykové verze, ze které byla provedena extrakce dat. Důvodem pro vytvoření více databází byla nejednotnost URI při extrakci z různých jazykových verzí Wikipedie. Každá jazyková verze totiž zcela přirozeně obsahuje mnohem více obsahu k tématům, které se týkají oblastí, ve kterých je daný konkrétní jazyk významně používán (např. volby ve Francii, obce v Německu, atp.). Důsledkem bylo, že data extrahovaná z anglické verze Wikipedie, kterých byla většina, měla jako subjekt URI obsahující název článku v angličtině, a data extrahovaná z jiných jazykových verzí, měla jako subjekt URI obsahující název článku v daném jazyce.

Verze DBpedia 2014 se skládá ze 125 lokálních verzí, zároveň však stále existuje tzv. kanonická databáze, ve které jsou obsažena všechna data extrahovaná z anglické verze Wikipedie, a navíc názvy a abstrakty v ostatních jazycích, pokud pro ně existuje ekvivalentní článek jak v anglické, tak v odpovídající cizojazyčné verzi Wikipedie. Dále v této práci bude pracováno právě s touto kanonickou databází, nebude-li uvedeno jinak. Všechny verze dohromady popisují 38,3 mil. „věcí“ v cca 3 mld. RDF trojic. Odděleně od těchto datasetů existuje ještě verze *DBpedia Live*, která je neustále, „on-the-fly“, aktualizována podle posledních změn anglické verze Wikipedie. Pro tento účel existuje upravená verze programu *DBpedia Information Extraction Framework*, napsaná v jazyce Java, která nepřetržitě extrahuje data ze změněných článků, nahrává změny do samostatného SPARQL endpointu[1] a vytváří speciální změnové soubory v RDF serializaci N-Triples, s frekvencí přibližně 1 soubor za sekundu.

DBpedia používá pro popis své ontologie jednak vlastností zavedených RDF slovníků, jako jsou FOAF[2] nebo PROV[3], tedy např. vlastnosti `foaf:depiction`[4] nebo `prov:wasDerivedFrom`[5], jednak definuje vlastnosti svoje vlastní, např. `dbp:reference`[6]. Zeměpisné souřadnice jsou v ontologii popsány převážně pomocí standardizovaných slovníků *Basic Geo (WGS84 lat/long) Vocabulary*[7] a *GeoRSS Simple encoding of the W3C Geospatial Vocabulary*[8]. [11] Obdobně jsou v ontologii DBpedia používány implicitní datové typy z formátů XML a RDF, např. `rdf:langString`, `xsd:anyURI`[9], `xsd:string` nebo `xsd:time`, ale především je zavedeno velké množství vlastních datových typů.

[1] http://live.dbpedia.org/sparql
[2] http://www.foaf-project.org/
[3] https://www.w3.org/ns/prov
[4] foaf: <http://xmlns.com/foaf/0.1/>
[5] prov: <http://www.w3.org/ns/prov#>
[6] dbp: <http://dbpedia.org/resource/>
[7] http://www.w3.org/2003/01/geo/
[8] http://www.w3.org/2005/Incubator/geo/XGR-geo/
[9] xsd: <http://www.w3.org/2001/XMLSchema#>

1.3.2 Naplňování databáze DBpedia

Informace z Wikipedie jsou převáděny do databáze DBpedia pomocí extrakčních algoritmů, napsaných v jazyce PHP a fungujících jako součást frameworku *DBpedia Information Extraction Framework*, který je dostupný pod svobodnou licencí GNU GPL v2 na serveru GitHub[1]. Software MediaWiki, na kterém je Wikipedie provozována (a který je též napsaný z větší části v PHP), označuje některé informace obsažené v článcích na Wikipedii speciálními syntaktickými značkami. Podle těchto značek lze text procházet a uspořádávat takto označené informace do struktury. [4, 22]

Kromě těchto značek obsahuje Wikipedie též částečně strukturovaná data v tzv. MediaWiki šablonách, jako jsou Infoboxy, Navboxy, Rozcestníky, ale i tabulky nebo seznamy. Všechny tyto jednotlivé prvky lze procházet automatizovaným algoritmem a vytvářet na jejich základě RDF trojice. [4]

Ve většině případů se subjektem RDF trojice stává IRI adresa odpovídajícího konceptu v databázi DBpedia. IRI adresa ve znalostní bázi DBpedia vznikne spojením URL `http://dbpedia.org/resource/` a názvu článku ve Wikipedii. Např. tedy `http://dbpedia.org/resource/Plzeň`. Algoritmus poté hledá počátek šablony v syntaxi MediaWiki (znaky „`{{`“) a její atributy převádí na predikáty. Hodnoty atributů se převedou na objekty a RDF trojice je tak kompletní. [4, 5] Podrobnosti a problémy spojené s analyzováním MediaWiki šablon lze nalézt např. v [5]. Infobox Wikipedie pro článek Plzeň je na obrázku 2a a jeho RDF reprezentaci ilustruje obrázek 2b.

1.3.3 Přístup k datům projektu DBpedia

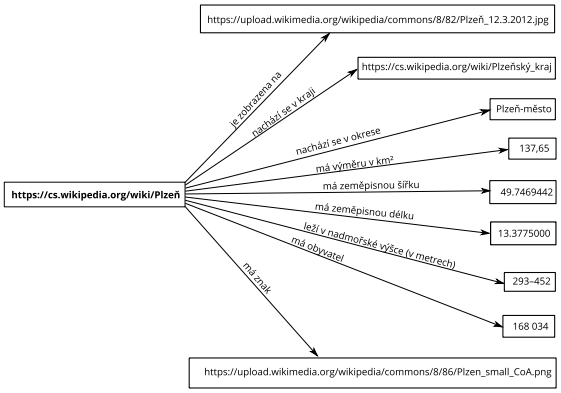
Na data uložená v ontologii DBpedia se lze dotazovat pomocí SPARQL endpointu[2] nebo je lze procházet jako Linked Data Interface pomocí běžného prohlížeče webových stránek. V roce 2015 se webové rozhraní pro procházení dat změnilo na graficky i uživatelsky přívětivější verzi, zvanou DBpedia Viewer. [15]

Data jsou dostupná ke stažení buď jako RDF serializace ve formátech Turtle, N-Triples nebo N-Quads. [4] Na webu Univerzity v Mannheimu[3] jsou data dostupná navíc i v serializaci Turtle Quads, která je obdobou N-Quads, a která byla navržena přímo pro potřeby projektu DBpedia.[4] N-Triples a N-Quads používají pro dereferenci prvků URI, zatímco Turtle a Turtle Quads používají IRI. Tedy např. zatímco v N-Triples bude instance popisující Plzeň odkazována jako `<http://dbpedia.org/resource/Plze%C3%AD>`,

[1] `https://github.com/dbpedia/extraction-framework/wiki`
[2] `http://dbpedia.org/sparql`
[3] `http://data.dws.informatik.uni-mannheim.de/dbpedia/2014/`
[4] `http://rdfpro.fbk.eu/tql.html`

```
 {{Infobox statutární město
| název = Plzeň
| foto = File:Plzeň 12.3.2012.jpg
| popisek.foto = Katedrála sv.
Bartoloměje na náměstí Republiky
| znak = Image:Plzen small CoA.png
| článek o znaku = Plzeňský znak
| vlajka = Soubor:Flag of Plzen.svg
| kraj = [[Plzeňský kraj|Plzeňský]]
| okres = Plzeň-město
| země = [[Čechy]]
| nad.výš = 293-452
| výměra = 137,65
| obyvatelé = 168 034
| zeměpisná šířka = 49.7469442
| zeměpisná délka = 13.3775000
}}
```

(a) Podoba strukturovaných dat v Infoboxu v článku „Plzeň“ na české Wikipedii.



(b) Reprezentace dat extrahovaných z Infoboxu na Wikipedii do formátu RDF.

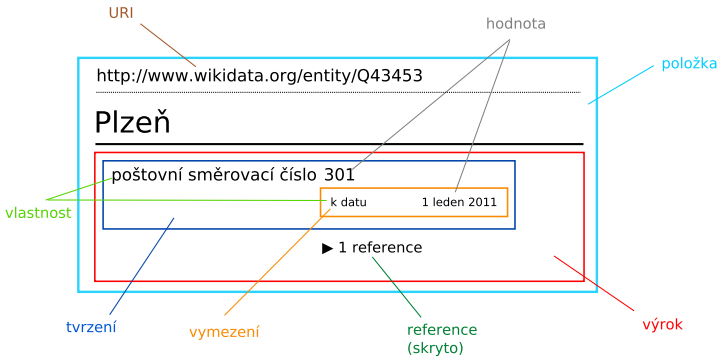
Obrázek 2: Podoba Infoboxu k článku Plzeň v syntaxi MediaWiki a jeho reprezentace pomocí RDF trojic.

v Turtle bude mít odkaz podobu `<http://dbpedia.org/resource/Plzeň>`. Rovněž lze stáhnout ontologii, obsahující pouze řízený slovník tříd a vlastností ve formátu OWL, v serializacích OWL/XML a N-Triples. Taktéž existuje zvláštní verze *DBpedia as Tables*, která je určená pro stažení ve formátech CSV a JSON.

1.4 Struktura databáze Wikidata

1.4.1 Datový model databáze Wikidata

Data v projektu Wikidata mají podobu tzv. *položek*. Každá *položka* (angl. *item*) je jednoznačně identifikována svým URI a může k sobě obsahovat jisté *výroky* (angl. *statements*), a seznam stránek v projektech WMF, propojených s touto položkou. Výrok sestává ze dvou částí: *tvrzení* (angl. *claim*) a reference k danému *tvrzení*. *Tvrzení* je složeno opět ze dvou základních částí: *Vlastnosti* (angl. *property*) a hodnoty dané *vlastnosti*. *Vlastnost* je stránka na webu Wikidata s jednoznačnou URL, oproti *položkám* však nemůže obsahovat seznam propojených stránek. Nad tento rámec může být součástí *tvrzení* ještě tzv. *vymezení* (angl. *qualifier*), které dále upřesňuje, k čemu se vztahuje hodnota *tvrzení* nebo dává jiné doplňující informace. Přehledněji celou strukturu zobrazuje obrázek 3.



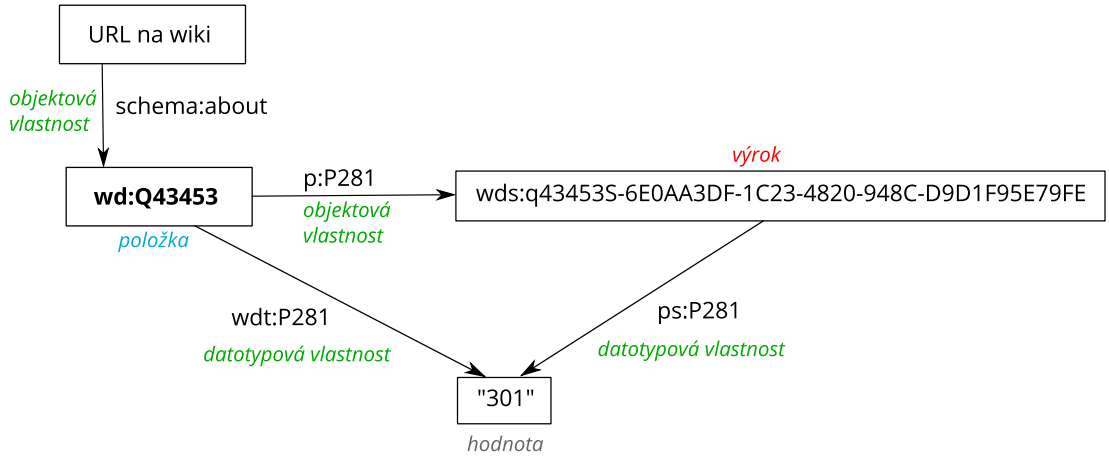
Obrázek 3: Struktura položky databáze Wikidata. (Upraveno podle Tchoř, CC-BY-SA 3.0)

Projekt Wikidata se snaží být maximálně na jazyce nezávislým. *Položka* tak není určena svým názvem, nýbrž svým ID, které se skládá z písmene „Q“ a pseudonáhodně určeného čísla. Toto ID je pak součástí URL, které *položku* definuje i z pohledu standardu RDF. Název *položky*, nazývaný *štítek*, lze pak nastavit v každém jazyce zvlášť, stejně jako její stručný popis a tzv. aliasy. V [6] je uvedeno, že jazyků projekt Wikidata podporuje 358. *Vlastnosti* projektu Wikidata jsou taktéž samostatné stránky s vlastním URL, začínají však vždy písmenem „P“. Každá vlastnost smí nabývat jen určitých předem daných hodnot. Tato omezení vznikají přiřazením datových typů, kterých v současnosti Wikibase DataModel rozpoznává celkem 11. *Položky* a *vlastnosti* lze souhrnně označit jako *entity*. Rozšíření MediaWiki, které umožňuje takovouto práci s daty se nazývá Wikibase a celý datový model nese název Wikibase DataModel.

Jelikož datový model Wikibase DataModel má odlišnou sémantiku a komplexnější syntaxi, než jakou je možné popsat pomocí RDF, musí vždy při vytváření nového RDF dumpu dojít k namapování potřebných tříd v kódu MediaWiki do struktury RDF. V [10] je popsán základní postup této konverze do ontologie kompatibilní se standardem OWL 2. Od roku 2014 se však způsob konverze poněkud změnil, především pokud jde o URI jednotlivých prvků modelu.[1] Základem je tzv. konkretizace (angl. reification). Ta zahrnuje označení *výroků*, *referencí* a složených datových typů samostatným URI, vytvořeným hashovací funkcí. Tyto celky se tak stanou odkazovatelnými z jiných částí grafu. V RDF reprezentaci je potom *položka* subjektem a celý *výrok* objektem. Zároveň však je *položka* propojena i přímo s hodnotou dané vlastnosti, pro jednoduché případy, kdy není potřeba přistupovat k *vymezením* a k *referencím*. Jako zjednodušený příklad bez *vymezení* i *referencí* je na obrázku 4 zobrazena *položka* „Plzeň“ (Q43453) s *vlastností* „poštovní směrovací číslo“ (P281), namapovaná do RDF. Obrázek 4 vznikl

[1] https://lists.wikimedia.org/pipermail/wikidata/2016-May/008635.html

podle schématu na webu Wikimedia Commons[1]. Celý export je prováděn knihovnou Wikidata Toolkit[2], napsanou v jazyce Java.



Obrázek 4: Detail RDF reprezentace konkretizovaného výroku s vlastností P281 (poštovní směrovací číslo). Zkratky jmenných prostorů jsou následující:

- `schema:` <http://schema.org/>
- `wd:` <http://www.wikidata.org/entity/>
- `wdt:` <http://www.wikidata.org/prop/direct/>
- `p:` <http://www.wikidata.org/prop/>
- `wds:` <http://www.wikidata.org/entity/statement/>
- `ps:` <http://www.wikidata.org/prop/statement/> .

1.4.2 Naplňování databáze Wikidata

Po svém spuštění v roce 2012 začala být Wikidata roboticky (automatickým, či poloautomatickým algoritmem) naplňována podle článků v různých jazykových verzích Wikipedie. Toto probíhalo prakticky tak, že ke každému článku na Wikipedii (primárně té anglické) byla vytvořena *položka*, která byla poté opět roboticky doplněna o odkazy do dalších jazykových verzí Wikipedie, a o příslušné *štítky*. [6] Ve druhé fázi naplňování byla k položkám přidávána *tvrzení* kopírováním informací ze šablon v příslušných článcích na Wikipedii, primárně z Infoboxů. Jako *reference* k danému *tvrzení* pak bylo vyplněno např. „převzato z: česká Wikipedie“, aby bylo možné dohledat původní zdroj informace.

V současnosti obsahuje databáze Wikidata více než 17,5 mil. *položek*. Cílem projektu Wikidata není pojmout a schraňovat všechna existující data, ale sloužit jako

[1] Smalyshev (WMF), CC0 1.0, https://commons.wikimedia.org/wiki/File:Rdf_mapping.svg
[2] https://github.com/Wikidata/Wikidata-Toolkit

tzv. sekundární databáze – shromažďovat již dříve zveřejněná fakta a odkazovat se na jejich zdroje.

1.4.3 Přístup k datům projektu Wikidata

Data jsou pravidelně exportována do tzv. dumpů, primárně ve formátu JSON. Dump je obecně používaný výraz, označující zálohu či výstup z databáze určený k přesunu dat[1]; do češtiny se obvykle nepřekládá. Dostupné jsou i verze dumpů ve formátu XML a RDF-kompatibilní verze ve formátu N-Triples. [6] Zásadní nevýhodou výstupních formátů XML a JSON je, že nejde o žádnou z RDF-kompatibilních serializací (RDF/XML, JSON-LD, ...), nýbrž o čistě XML dokument s vlastními definovanými elementy, resp. obecný JSON dokument.

Pro jednoduchý a omezený přístup k datům slouží veřejně přístupné API[2], které je v neustálém vývoji a podpora komplexnějších dotazů, jako „Kolik vnuků Eduarda VII. se dožilo alespoň 50 let“ nebo „Kteří spisovatelé se narodili v Plzni“, je zatím hudbou budoucnosti. Pro dotazy lze zatím využít některého z externích nástrojů na serveru WMF Labs[3], nejčastěji *WikidataQuery API*, který umožňuje např. i vyhledávání položek podle polohy pomocí příkazu AROUND (položky v určitém okolí od daných souřadnic). Ukázka webové aplikace, využívající WikidataQuery API a vyhledávání pomocí polohy je v příloze C.

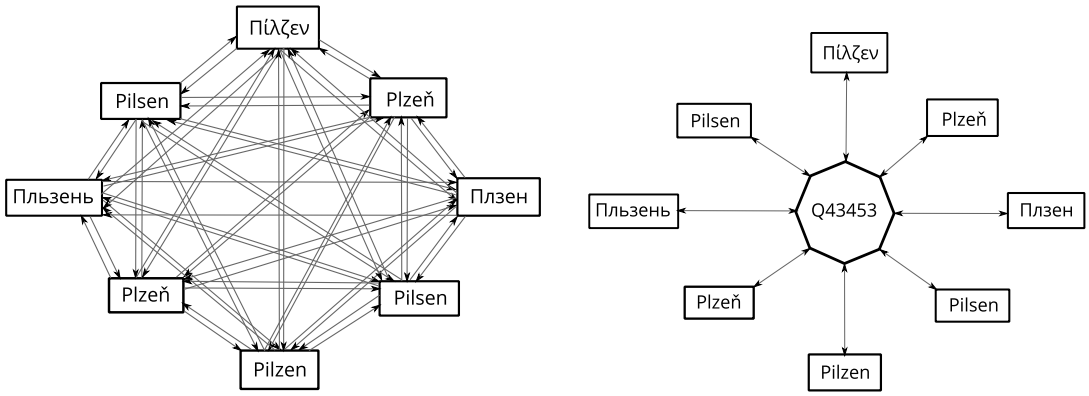
V březnu 2015 byl zprovozněn první testovací SPARQL endpoint určený výhradně pro Wikidata.[4] Již dříve byly RDF dumpy z projektu Wikidata nahrány do rozsáhlé databáze spravované společností OpenLink Virtuoso[5], která zpřístupňuje značnou část LOD cloudu. V září 2015 byl spuštěn oficiální SPARQL endpoint spravovaný WMF[6], pojmenovaný *Wikidata Query Service*[7].

1.4.4 Využití databáze Wikidata v rámci Wikipedie

Prozatím nejvýznamnější změnou je změna systému odkazování na jiné jazykové verze Wikipedie, tzv. interwiki odkazy. Interwiki odkazy musely být součástí každé jazykové verze článku, pojednávajícím o daném tématu a to navíc v podstatě duplicitně. Například český článek Plzeň musel obsahovat interwiki odkaz na německý článek Pilsen,

[1] http://www.oxforddictionaries.com/definition/english/data-dump
[2] https://www.wikidata.org/w/api.php
[3] https://tools.wmflabs.org/
[4] https://lists.wikimedia.org/pipermail/wikidata-l/2015-March/005683.html
[5] http://lod.openlinksw.com/sparql
[6] https://lists.wikimedia.org/pipermail/wikidata/2015-September/007042.html
[7] https://query.wikidata.org

ale německý článek Pilsen musel taktéž obsahovat interwiki odkaz zpět na český článek Plzeň. Po spuštění projektu Wikidata jsou všechny články na každé jazykové mutaci Wikipedie propojeny se svou odpovídající datovou *položkou* v databázi Wikidata a propojení mezi ostatními mutacemi je zajištěno skrze tuto *položku*. Ekvivalentně funguje propojení i mezi jinými projekty WMF (Wikizdroje, Wikislovník, ...). Tuto změnu, někdy nazývanou jako *Fáze 1*, ilustruje na zjednodušeném příkladu Plzně obrázek 5.



Obrázek 5: Schéma vzájemného propojení článků Wikipedie před zavedením databáze Wikidata (vlevo), a po jejím spuštění.

Pro tuto práci mnohem zajímavější změna je však přejímání *tvrzení* z projektu Wikidata do Infoboxů. Prozatím je potřeba mluvit jen o *tvrzeních*, neboť systém referencí projektu Wikidata není ještě v praxi zcela používaným, a tak Wikipedie prozatím nepřebírá celé *výroky* včetně referencí. Některé Infoboxy jsou již alespoň částečně generovány automaticky a jejich obsah není vyplněn v šabloně v článku, ale je přebírán z odpovídající datové *položky* v databázi Wikidata. Tato změna bývá někdy označována jako *Fáze 2*.

Prozatím stále v očekávání je *Fáze 3*, která umožní komplexní dotazování do databáze Wikidata, což pro Wikipedii představuje především automatizované vytváření seznamů.

1.5 DBpedia a Wikidata v Linked Open Data síti

Vzhledem k principům, popsaným v kapitolách 1.3.2 a 1.4.4, by se mohlo zdát, že postup přebírání informací vypadá přibližně jako na obrázku 6. Tak tomu ale není. Data, která jsou do Wikipedie přebírána z databáze Wikidata, již totiž nejsou přímo v kódu Wikipedie uvedena, a pakliže nejsou v šablonách MediaWiki, extrakční algoritmus je nemůže najít a převést do databáze DBpedia. Poněkud paradoxně tak Wikidata vznikají na úkor projektu DBpedia. Neznamená to ale, že by rázem přestalo být možné data

z Wikipedie do znalostní báze DBpedia extrahovat. Při automatickém přenášení dat z Wikipedie do databáze Wikidata totiž dojde jen ke zkopírování a informace je poté duplicitně uvedena jak v projektu Wikidata, tak i v šabloně na Wikipedii. Až po ruční kontrole editory Wikipedie jsou postupně některé duplicity z Wikipedie odmazávány ve prospěch zobrazení dat z projektu Wikidata. Extrakce dat do databáze DBpedia je tedy stále možná, ale rozsah dostupných dat je postupně omezován.

Řešením pro databázi DBpedia, jak nepřicházet o data přesunutá na jiné místo, je jejich následování. Jak již bylo zmíněno v kapitole 1.3.1, v roce 2015 byl *DBpedia Information Extraction Framework* rozšířen o extrakční algoritmus z projektu Wikidata. Vznikla tak databáze *DBpedia Wikidata*[1], která obsahuje všechna data z projektu Wikidata (v době extrakce), popsaná ontologií DBpedia. [22]

(a) Data ve webovém rozhraní projektu Wikidata.

(b) Data zobrazená v Infoboxu Wikipedie.

(c) Data ve webovém rozhraní projektu DBpedia.

Obrázek 6: Různé vyjádření stejných dat v projektech Wikidata, Wikipedia a DBpedia.

Wikidata obsahují dvě vlastnosti, provazující položky databáze s prvky OpenStreetMap. První z nich je „ID relace OpenStreetMap“ (P402), a druhá „značka či klíč na OpenStreetMap“ (P1282). Pomocí SPARQL dotazu bylo zjištěno, že položek, které obsahují alespoň jednu z těchto vlastností je dohromady 17 596, z čehož 16 900 tvoří P402.

Naopak, do atributů prvků v projektu OpenStreetMap začaly být od počátku roku 2015 přidávány odkazy na položky v databázi Wikidata. V květnu 2015 existovalo 32 681 takto propojených prvků, z čehož 24 259 odkazů obsahovaly *relace* OpenStreetMap, zbylých 8 422 odkazů obsahovaly *uzly* a *cesty*. [7]

[1] http://wikidata.dbpedia.org/

Podobný postup jako při tvorbě znalostní báze DBpedia použili její autoři také při tvorbě databáze LinkedGeoData[1]. Ta používá extrakční algoritmy k namapování objektů projektu OpenStreetMap do jazyka RDF. Výsledná datová báze je dostupná ve formě N-Triples dumpů, přes REST API a skrze SPARQL endpoint[2]. [25]

Databáze Freebase, která vznikla v roce 2007 pod hlavičkou firmy Metaweb, byla až do nedávna klíčovou součástí znalostní báze *Knowledge Graph*, firmy Google. Díky Freebase, resp. Knowledge Graph, může vyhledávač Google fungovat jako odpovídací stroj (angl. answering engine), obdobně jako Wolfram|Alpha[3] nebo Ask.com[4]. Je vhodné věnovat zde Freebase alespoň krátkou zmínku, neboť ji lze považovat z části za logického předchůdce projektu Wikidata. Obsah databáze vznikl přebíráním informací ze zdrojů jako Wikipedie, NNDB[5] nebo MusicBrainz[6], ale taktéž bylo uživatelům umožněno přidávat vlastní tvrzení. Obsahuje popis celkem 47 560 817 konceptů, zvaných *témata* (angl. *topics*).

V systému Wikidata existuje vlastnost P646 – *identifikátor Freebase* (angl. *Freebase identifier*), která propojuje položky s tématy na Freebase. Pomocí SPARQL dotazu bylo zjištěno, že položek s vlastností P646 je celkem 1 153 726. V prosinci 2014 firma Google oznámila, že provoz Freebase bude postupně utlumován a obsah databáze bude migrován na Wikidata. Poté, co několik měsíců API Freebase fungovalo v režimu jen pro čtení, byl v květnu 2016 ukončen jeho provoz úplně a na webu jsou dostupné jen kompletní dumpy dat ke stažení.[7] Migrace dat z Freebase prozatím rozšířila databázi Wikidata o 14 mil. tvrzení. Podrobnosti importu jsou popsány v [27].

[1] http://linkedgeodata.org/About
[2] http://linkedgeodata.org/sparql
[3] http://www.wolframalpha.com/
[4] http://www.ask.com/
[5] http://www.nndb.com/
[6] https://musicbrainz.org/
[7] https://lists.wikimedia.org/pipermail/wikidata/2016-May/008664.html

2 Metodika srovnávání znalostních bází

2.1 Diskuse existujících řešení srovnávání znalostních bází

Ačkoliv se ontologie, a znalostní báze obecně, v posledních letech těší velkému rozmachu a míra jejich využití pro různé aplikace roste (někdy až nadměrně), jen málo odborných prací bylo prozatím publikováno, které by se věnovaly jejich srovnávání. V [12] je uvedena metodika srovnávání sémantických vazeb v ontologiích, která je poté ověřena na 3 „referenčních ontologiích“. Naopak v [13] je popsán zcela konkrétní případ porovnání 4 existujících právních ontologií, za účelem vytvoření „právního znalostního systému“ a příspěvek [16] na konferenci *Linked Data on the Web 2015* je zaměřen na automatickou klasifikaci LOD datasetů do tematických tříd, pomocí četnosti použitých vlastností a slovníků.

Některé práce se věnují srovnávání ontologických reasonerů. Korejský výzkum [20] srovnává 4 nejčastěji používané reasonery na základě rychlosti vyhodnocování dotazů. Americký článek [19] porovnává tytéž reasonery (a jeden další), přičemž se soustřeďuje na to, jak je výpočetní čas dotazu ovlivňován kladením dodatečných uživatelských pravidel. Ismail a Bakar se ve své práci ([21]) věnují srovnání automatického a manuálního způsobu vytváření ontologií.

Nejkomplexnější metodu srovnávání ontologií představili Alasoud et al. v [18]. Jimi navržený „multi-level matching algorithm“ hledá navzájem si odpovídající pojmy ve dvou porovnávaných ontologiích. Toto hledání sestává z dvojího porovnání pojmů v ontologiích jednak podle syntaktické blízkosti slov, jednak podle jejich sémantické blízkosti s pomocí ontologického slovníku WordNet[1]. Výsledkem srovnání je binární shodnostní matice Map_{0-1}. Takový postup je exaktní, ale také velmi časově i výpočetně náročný a tudíž pro potřeby této práce nadměrně komplikovaný. Jelikož se tato práce zaměřuje na srovnání jen určité části ontologií – prostorových dat – je s použitím zmíněných zdrojů v následující kapitole navržena jednodušší a specifičtější metoda srovnání.

2.2 Navržení metody pro porovnání projektů DBpedia a Wikidata

Hovy ve svém článku ([12]) doporučuje rozčlenit porovnání ontologií na 3 fáze:

1. základní charakteristika (angl. *general characteristics*),
2. globální rozdíly (angl. *overall differences*),

[1] http://wordnet.princeton.edu/

3. vybrané odlišnosti (angl. *points of difference*).

Základní charakteristika obou řešených znalostních bází již byla popsána v kapitolách 1.3 a 1.4, proto se jí již nebudeme dále zabývat. Body (2) a (3) je potřeba přizpůsobit na téma prostorových dat a jejich využití.

ad 2) Hovy dále navrhuje rozdělit globální rozdíly podle *a*) formy, *b*) obsahu a *c*) použití.

a) Globální rozdíly ve formě budou dále nazývány jako rozdíly „obecných parametrů“, a tvoří **první část** srovnání uvedeného v následující kapitole. Jednotlivé parametry pro porovnání byly voleny s ohledem na cíle práce a podle [12] a [17]. Jde např. o srovnání *vícejazyčnosti*, *datových typů* nebo *metod vzniku databází*

b) Globální rozdíly v obsahu porovnávány nebudou, neboť by takové porovnání bylo vzhledem k širokému („cross-domain“) zaměření obou znalostních bází velmi rozsáhlé a šlo by zcela nad zamýšlený rámec této práce.

c) Globální rozdíly v použití jsou naopak pro tuto práci zásadní. Jejich porovnání je **druhou částí** použité srovnávací metody. Parametry tohoto porovnání vycházejí z cíle porovnat báze Wikidata a DBpedia jako *zdroje prostorových dat*. Rozdíly v API, prohlížecích nástrojích, licenci dat, aj. uvádí jednak [12], jednak se na ně adekvátně zaměřují i další práce jako např. [19].

ad 3) Vybrané odlišnosti zde budou omezeny pouze na prostorové vlastnosti dat. Vytvoření metody porovnání prostorové složky dat vychází z vlastností prostorových dat, uvedených zejména v [14] a [17], přičemž toto porovnání tvoří **třetí část** následně použité srovnávací metody. Vzhledem ke struktuře databází je logické rozdělit tuto část na *a*) srovnání geometrického a polohového určení prvků, což zjednodušeně řečeno znamená porovnání objektů se souřadnicemi, a dále na *b*) srovnání vyjádření prostorových vztahů mezi objekty.

ad a) Kritéria porovnání geometrického a polohového určení jsou výběrem a agregací vlastností uvedených v [14] a v [17], relevantních pro srovnávané databáze. Konkrétně jde o *popis geometrie*, *popis souřadnic*, *podporu různých souřadnicových systémů* a *podrobnost souřadnic*. Poslední bod je záměrně pojmenován *podrobnost*, nikoliv *přesnost*, protože nejde o posouzení správnosti určení polohy, ale o prostorové rozlišení. Posouzení přesnosti souřadnic v řešených databázích by bylo komplikované, časově náročné, a vystačilo by na samostatnou kvalifikační práci.

ad b) Výběr kritérií porovnání prostorových vztahů ve struktuře ontologií je převzat z přehledu prostorových vztahů ve 2D prostoru, uvedeném v [23]. Zde uvedený model byl vybrán, protože je poměrně jednoduchý (jen 8 zobecněných vztahů), obecně aplikovatelný, široce respektovaný a později se stal základem pro standardizovaný model DE-9IM. Pro potřeby porovnání v této práci byly navíc sloučeny vztahy „obsahuje“ a „je obsažen v“ i „překrývá“ a „je překryt“. To je možné, protože uvedené vztahy jsou vzájemně inverzní, a ontologické reasonery mohou z jednoho vztahu odvodit druhý.

Nad tento rámec je do metody porovnání přidána ještě **část čtvrtá**, která zachycuje kvantitativní vlastnosti obou znalostních bází ve stejném čase. Tato kritéria jsou z předchozích skupin vyčleněna, neboť jejich hodnoty se v čase rychle mění.

3 Porovnání projektů DBpedia a Wikidata

V následující kapitole je provedeno srovnání projektů Wikidata a DBpedia podle metody navržené v části 2.2. Každá podkapitola se věnuje jedné skupině porovnávaných parametrů, a je členěna následovně: každý parametr je nejprve podrobně rozebrán a v závěru každé podkapitoly je uvedena shrnující tabulka.

3.1 Porovnání obecných parametrů databází

- *Vícejazyčnost* – Obě znalostní báze jsou navrženy tak, aby umožnily ukládat informace v co možná největším množství jazyků. To vyplývá i z mnohojazyčné povahy Wikipedie, která je spojovacím článkem obou databází. Zatímco DBpedia se vydala směrem vydávání jazykově oddělených datasetů, Wikidata jsou od počátku modelována jako jednotná, jazykově nezávislá databáze, s možností zobrazení popisů prvků jen ve vybraném jazyce (podrobněji bylo popsáno v kapitole 1.4). DBpedia byla takto jednotná až do verze 3.6, poté ale byl přístup změněn, kvůli vzájemné neslučitelnosti jazykových verzí Wikipedie (podrobněji bylo popsáno v kapitole 1.3). K jednotnému přístupu se DBpedia pravděpodobně v budoucnu opět vrátí, což je umožněno právě zavedením projektu Wikidata a extrakcí jeho obsahu do ontologie DBpedia (podrobně viz 1.5). [22]

- *5hvězdičková stupnice přístupnosti dat* – Na 5hvězdičkové stupnici přístupnosti, obě databáze bezpochyby spadají do nejvyšší kategorie. Oba projekty jsou veřejně dostupné, se strojově čitelnými daty ve standardizovaných formátech, s dereferencovatelnými objekty a s bohatým propojením s dalšími databázemi, i mezi sebou. Spekulativní může být splnění třetího kritéria (standardizovaný formát dat) u databáze Wikidata. Ačkoliv vnitřně mají Wikidata strukturu svou vlastní, všemi přístupovými body lze získat data ve standardizovaných formátech jako je RDF serializace N-Triples nebo JSON.

- *Datové typy* – V současnosti Wikibase DataModel implementuje celkem 11 datových typů. Jde o: *média na Commons*, *zeměpisné souřadnice*, *položka*, *vlastnost*, *řetězec*, *jednojazyčný text*, *množství*, *čas*, *externí identifikátor*, *matematický výraz* a *URL*.[1]. Další datové typy, jako *geografický tvar* nebo *vícejazyčný text* jsou v návrhu.[2] Naopak databáze DBpedia má zavedeno celkem 381 datový typ. To je způsobeno tím, že ve znalostní bázi DBpedia je zaveden datový typ pro každou použitou měrnou jednotku, tj. např. *BelarussianRuble* pro běloruský rubl, *Day*

[1] https://www.wikidata.org/wiki/Special:ListDatatypes
[2] https://www.wikidata.org/wiki/Wikidata:Development_plan#Todo

pro den, *Gigametre* pro v reálu nepoužívanou jednotku SI gigametr, *Minute* pro libovolné minuty, ale též *Speed* pro vyjádření rychlosti v libovolných jednotkách, či *Currency* pro libovolnou měnu, a taktéž *rdf:langString*[1] pro jazykově rozlišený text, či *xsd:float* pro libovolné číslo s desetinným rozvojem.[2]

- *Ontologické vlastnosti* – Rozšířená verze definice ontologie podle N. W. Borsta ([24]) zní „formální specifikace sdílené konceptualizace". Nazývat některou z řešených databází, jako celek, ontologií není obvyklé, neboť data uložená ve fyzickém datovém modelu jsou popsána s jinou úrovní abstrakce, než v konceptuálním modelu. Specifikací konceptualizace je tak spíše samotný datový model těchto databází. I proto jsou ontologií nazývány jen uzavřené podmnožiny obou databází. V případě znalostní báze DBpedia jde o taxonomii vlastností a tříd ve jmenném prostoru `http://dbpedia.org/ontology/`, v případě projektu Wikidata jde o řízený slovník vlastností a tříd definujících datové typy, umístěný na adrese `http://wikiba.se/ontology#`. Specifikace je v obou případech formální, neboť obě ontologie jsou popsány formálním jazykem OWL.
  - *Rozlišení tříd a individuálů* – Projekt DBpedia rozlišení na třídy a individuály důsledně dodržuje. Verze DBpedia 2014 obsahuje 746 tříd a podtříd uspořádaných do stromové struktury[3]. Databáze Wikidata naopak toto dělení v základu neimplementuje. Strukturu tříd, podtříd a individuálů je ale možné rekonstruovat díky vlastnostem „instance (čeho)" (P31) a „nadtřída" (P279).
  - *Restrikce* – Při vytváření struktury projektu Wikidata nebyly restrikce ani pravidla nijak řešeny. Až při praktickém používání dat a údržbě znalostní báze jako celku, se ukázalo zavedení restrikcí jako nevyhnutelné. To bylo vyřešeno tak, že na diskusní stránce, která existuje pro každou vlastnost, může být uvedeno, zda jde o vlastnost funkcionální, inverzně funkcionální, tranzitivní, symetrickou či antisymetrickou. Dále může být uveden seznam konfliktních vlastností, výčet předpokládaných hodnot nebo naopak neočekávaných hodnot, konfliktní či očekávaná vymezení, aj. Tyto definice jsou však jen neformálním textem, doprovázejícím dané vlastnosti, a dodržování těchto restrikcí není při editaci dat striktně vyžadováno. V případě DBpedia jsou restrikce řešeny jen u vlastností uvedených v ontologii DBpedia. Vlastnosti

[1] rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
[2] http://mappings.dbpedia.org/index.php/DBpedia_Datatypes#XML_and_RDF_Datatypes
[3] http://mappings.dbpedia.org/server/ontology/classes/

zde definované většinou obsahují vlastnosti `rdfs:range`[1] a `rdfs:domain`, které určují rozsah objektu valstnosti, resp. rozsah subjektu.

- *Využití slovníků třetích stran* – Jelikož databáze Wikidata má svou vlastní vnitřní RDF-nekompatibilní strukturu, která vychází především ze softwaru MediaWiki, využívá pro popis objektů vlastních datových typů i vlastností. Ve verzi exportované do RDF jsou používány některé základní vlastnosti ze standardu RDF a z vybraných slovníků, jako např. `schema:about`. Náznakem standardizace specifických vlastností je jejich namapování na vlastnosti v externích slovnících, pomocí vnitřní vlastnosti „shodná vlastnost“ (P1628). Ontologie DBpedia taktéž definuje svoje vlastnosti a datové typy, využívá ale široce i externích slovníků. Využívá především vlastností ze jmenných prostorů `rdfs:`[2], `owl:`[3], `dc:`[4] nebo `skos:`[5], přičemž někdy uvádí i více ekvivalentních vlastností z různých domén, včetně vlastního slovníku, zároveň.

- *Frekvence aktualizací* – Datasety DBpedia jsou aktualizovány nepravidelně, průměrně přibližně jednou až dvakrát ročně. Během tvorby této práce vyšly datové sady verze DBpedia 2015-04 a DBpedia 2015-10. [22] Zcela odlišně je udržován obsah databáze Wikidata, jejíž obsah je editován nepřetržitě tisícovkami uživatelů. Za stabilní verzi tak lze považovat dumpy, jejichž aktualizační frekvence se značně liší.

- *Metody použité pro získání dat* – Projekt DBpedia odvozuje všechny své datasety z Wikipedie, případně jiných projektů Nadace Wikimedia, pomocí extrakčních algoritmů, jak bylo podrobně popsáno v kapitole 1.3.2. Wikidata je dobrovolnický projekt, stejně jako ostatní projekty WMF nebo např. projekt OpenStreetMap. Každý uživatel může data doplňovat nebo upravovat, dokonce bez nutnosti registrace. Všechny vlastnosti položek však musí být převzaty z jiného zdroje a opatřeny referencí na původní zdroj. V počátku projektu byla data přebírána pomocí robotických účtů z ostatních projektů WMF, převážně z Wikipedie, tedy obdobně, jako v případě projektu DBpedia.

- *Ověřitelnost* – Věrohodnost a ověřitelnost dat v databázi Wikidata je teoreticky zaručena díky systému uvádění referencí. Prakticky však prozatím tento systém není zcela funkčním, neboť velké množství tvrzení je zcela bez reference a z těch

[1] rdfs: <http://www.w3.org/2000/01/rdf-schema#>
[2] rdfs: <http://www.w3.org/2000/01/rdf-schema#>
[3] owl: <http://www.w3.org/2002/07/owl#>
[4] dc: <http://purl.org/dc/elements/1.1/>
[5] skos: <http://www.w3.org/2004/02/skos/core#>

tvrzení, kde zdroj uveden je, je většina odkazem na jazykovou verzi Wikipedie, ze které byl údaj převzat. Což vzhledem k povaze Wikipedie nelze považovat za dostatečné. Ověřitelnost tak zvyšují odkazy do jiných databází, které pojednávají o stejném objektu, a kde je tak možnost správnost dat porovnat. Pro data v databázi DBpedia je pak tento způsob, tj. využití ostatních znalostních bází v LOD cloudu, jedinou možnou cestou ke strojovému ověření správnosti uváděných informací. Při přebírání dat z báze DBpedia je proto potřeba brát v potaz, že správnost uváděných informací přímo vyplývá ze správnosti informací ve Wikipedii, ze které jsou data převzata.

Tabulka 1: Porovnání obecných parametrů znalostních bází

Parametr	**DBpedia**	**Wikidata**
Multilingualita	Ano	Ano
5hvězdičková stupnice	5	5
Datové typy	381 specifických	11 obecných
Rozlišení tříd a individuálů	Ano	Možná rekonstrukce
Využití slovníků třetích stran	Částečně	Velmi omezeně
Restrikce	Pro některé vlastnosti	Pouze neformálně
Frekvence aktualizací	Průměrně $1-2\times$ ročně	Nepřetržitě
Metody použité pro získání dat	Extrakce	Sběr dat z jiných zdrojů
Ověřitelnost	Pomocí LOD odkazů	Pomocí referencí a LOD odkazů

3.2 Porovnání přístupnosti dat

- *Dumpy dat* – Pro projekt DBpedia je vydávání datasetů v dumpech jedním ze dvou klíčových přístupových bodů (spolu se SPARQL endpointem). Pro Wikidata jsou dumpy spíše doplňkovou možností k ostatním přístupovým bodům. Každopádně však oba projekty vycházejí ve formě stažitelných datových sad, ačkoliv v rozdílných formátech, a s rozdílnou pravidelností.

  – *Dostupné formáty* – Pro projekt Wikidata je primárně používaným formátem JSON, který je jednak implicitním formátem pro výstup API, jednak nejčastěji vydávaným formátem dumpů. Dále jsou exportovány stažitelné balíky ve formátech XML a RDF v serializaci N-Triples. Databáze DBpedia je dostupná v mnohem širší paletě formátů a zejména RDF serializací. Kromě JSON a N-Triples, jsou to ještě N-Quads, Turtle, Turtle Quads, OWL (v serializacích OWL/XML a N-Triples) a CSV, ačkoliv verze v JSON a CSV jsou speciálně upravené verze kanonické databáze s názvem *DBpedia*

as Tables. Doplňující informací v N-Quads a Turtle Quads oproti N-Triples a Turtle je URI článku na Wikipedii, z nějž byla informace extrahována. S tím souvisí i rozdíl verzí ve formátech Turtle a Turtle Quads, které používají IRI, oproti N-Triples a N-Quads, které používají URI (viz též 1.3.3). Ani jeden z projektů zatím nevydává dumpy celé databáze v nověji standardizovaném formátu JSON-LD, DBpedia jen umožňuje exportovat v něm jednotlivé objekty.

  – *Frekvence nových verzí* – Nové dumpy projektu DBpedia vycházejí vždy s novou verzí ontologie samotné. To se děje průměrně $1 - 2\times$ ročně. Databáze Wikidata vytváří dumpy pro každý formát zvlášť. Nejčastěji vychází verze ve formátu JSON, a sice $1\times$ týdně (každé pondělí). Dumpy v XML jsou vydávány jednou měsíčně, přičemž však lze stáhnout i „přírůstkové dumpy“, které obsahují jen změněné objekty od poslední verze dumpu, a které vycházejí denně. Nejnižší frekvenci pak má vydávání RDF N-Triples dumpů, které bylo poprvé uvedeno až v dubnu 2014 a od té doby probíhá nepravidelně, s odstupem mezi verzemi od 1 do 3 měsíců.

- *REST API* – Přístup k datům přes REST API je ze dvou řešených databází umožněn jen v projektu Wikidata. Standardní API softwaru MediaWiki je pro systém Wikibase rozšířeno o 23 specifických akcí, které umožňují přístup k položkám, vytváření výroků, slučování položek, přidávání a odebírání vymezení a referencí, aj. Toto API je neustále vyvíjeno a v budoucnu by mělo podporovat i takové dotazy, které by umožnily vyhledávání položek a seznamů položek obdobně jako při použití SPARQL endpointu. S projektem DBpedia sice souvisí *DBpedia Spotlight API*, jde ovšem oddělený projekt, jehož API neslouží k přístupu k datům, nýbrž k vyhledávání názvů konceptů databáze DBpedia v zadaném textu, a k jejich označení pomocí RDFa. DBpedia žádné REST API pro přímý přístup k datům nenabízí a ani jej není zapotřebí – prvky v ontologii DBpedia jsou mimo extrakci needitovatelné a přístup k datům v režimu čtení zajišťuje SPARQL endpoint.

  – *Podpora vyhledávacích dotazů* – Vyhledávací, jinak zvané „query“, dotazy přes REST API jsou pro projekt Wikidata velmi žádoucí, ale v současné verzi API nejsou možné. Z komunity Wikipedie existuje tlak na jejich implementaci, neboť jejich zavedení by umožnilo vytvářet automatické seznamy pro články Wikipedie přímo z dat projektu Wikidata (více viz kapitola 1.4.4). Prozatím takové dotazy podporují jen externí nástroje, které fungují jako nadstavba současného API, především jde o *WikidataQuery API*.

– *Podpora CORS – Cross-origin resource sharing* je systém umožňující přistupovat k datům z jiné domény, než je domovská (`http://www.wikidata.org/`), což není běžně možné kvůli bezpečnostnímu pravidlu Same-Origin Policy. To je pro využití dat v aplikacích třetích stran klíčové. API databáze Wikidata takový přístup umožňuje, ovšem jen pro předem definované domény, které jsou na seznamu ověřených domén.

- *SPARQL endpoint* – Nejjednodušším způsobem, jak vyhledávat v RDF databázích je dotazovací jazyk SPARQL. Rozhraní umožňující formulovat dotazy pro RDF data se nazývá SPARQL endpoint. Pro datovou bázi DBpedia je SPARQL endpoint primární způsob vzdáleného přístupu k datům. Její SPARQL endpoint[1] umožňuje přístup ke kanonické databázi a k 12 největším lokalizovaným datasetům.[2] DBpedia pro svůj SPARQL endpoint využívá software *Virtuoso Universal Engine*, který podle dokumentace[3] podporuje prostorové vyhledávání podle průniku prvků, vztahu „obsahuje“ a pomocí vzdálenosti. Funkce pro prostorové dotazy vycházejí ze standardu GeoSPARQL[4], ale jejich syntaxe se liší. Pro projekt Wikidata aktuálně funguje několik nezávislých SPARQL endpointů třetích stran a oficiální SPARQL endpoint *Wikidata Query Service*[5]. Nezávislé SPARQL endpointy mohou přistupovat k zastaralé nebo k výrazně zjednodušené verzi dat, oproti tomu služba *Wikidata Query Service* je aktualizována v čase blízkém reálnému a obsahuje téměř kompletní RDF export z databáze Wikidata (seznam výjimek oproti kompletním dumpům je dokumentován na webu MediaWiki.[6]) Provoz tohoto SPARQL endpointu zajišťují dva fyzické servery s grafovým databázovým prostředím *Blazegraph*.[7] Od května 2016 tento SPARQL endpoint podporuje i prostorové dotazy[8], avšak pomocí vlastních funkcí[9], nikoliv podle standardu GeoSPARQL.

- *Prohlížení webovým prohlížečem* – Přístup k datům srze rozhraní webového prohlížeče je pro Wikidata esenciální. Pomocí prohlížení HTML výstupu databáze mohou uživatelé data doplňovat a upravovat. Alternativní možností pro editaci

[1] http://dbpedia.org/sparql
[2] http://oldwiki.dbpedia.org/Permalink?page_id=435
[3] http://docs.openlinksw.com/virtuoso/rdfsparqlgeospat.html
[4] http://www.opengeospatial.org/standards/geosparql
[5] https://query.wikidata.org
[6] https://www.mediawiki.org/w/index.php?title=Wikibase/Indexing/RDF_Dump_Format&oldid=2109670#WDQS_data_differences
[7] https://lists.wikimedia.org/pipermail/wikidata/2016-February/008297.html
[8] https://lists.wikimedia.org/pipermail/wikidata/2016-May/008704.html
[9] https://www.mediawiki.org/wiki/Wikidata_query_service/User_Manual#Geospatial_search

je použití MediaWiki API s autorizovaným přístupem. DBpedia oproti tomu nabízí v HTML jen jednoduchou prohlížecí verzi, ačkoliv během roku 2015 bylo její webové rozhraní výrazně graficky modernizováno a uzpůsobeno k ručnímu procházení prvků DBpedia.

- *Licence a autorská práva* – Oba projekty jsou poskytovateli otevřených dat, což znamená, že jejich autoři se vzdávají části autorských práv. DBpedia nabízí svá data duálně pod licencemi *Creative Commons Attribution-ShareAlike 3.0 Unported License*[1] a *GNU Free Documentation License*[2]. To je přímým důsledkem přejímání dat z Wikipedie, která je distribuována pod stejnými licencemi, a z povahy těchto licencí, které vyžadují, aby každé odvozené dílo bylo šířeno za stejných podmínek jako dílo původní. Data z ontologie DBpedia lze tedy používat zcela volně, jedinými podmínkami je nutnost uvedení zdroje dat a šíření aplikovaného díla opět pod některou z licencí CC-BY-SA nebo GFDL. Wikidata jdou podstatně dálé. Každý uživatel při přispívání do databáze souhlasí se vzdáním se všech případných autorských práv ke vkládaným informacím a data jsou šířena pod licencí *Creative Commons CC0 1.0 Universal*[3]. To znamená, že je možné k nim přistupovat jako k volnému dílu (angl. public domain) a používat je libovolně a bez omezení.

Tabulka 2: Porovnání přístupnosti dat ze znalostních bází

Parametr	**DBpedia**	**Wikidata**
Dumpy dat	Ano	Ano
Formáty dumpů	`.json`, `.nt`, `.ttl`, `.nq`, `.owl`, `.nql`, `.csv`	`.json`, `.nt`, `.xml`
Frekvence nových dumpů	průměrně $1-2\times$ ročně	až $1\times$ týdně
REST API	Ne	Ano
API: Podpora vyhledávacích dotazů	–	Pomocí nástrojů 3. stran
API: Podpora CORS	–	Ano, s omezením
SPARQL endpoint	Ano	Ano
SPARQL: Podpora prostorových dotazů	Částečně	Částečně
Prohlížení webovým prohlížečem	Ano	Ano
Licence a autorská práva	CC-BY-SA & GFDL	CC-0

[1] http://creativecommons.org/licenses/by-sa/3.0/legalcode
[2] http://www.gnu.org/copyleft/fdl.html
[3] https://creativecommons.org/publicdomain/zero/1.0/legalcode

3.3 Porovnání prostorových vlastností dat

3.3.1 Geometrické a polohové určení objektů

- *Geometrie* – Geometrická reprezentace prvků obou znalostních bází je značně omezená. Dá se říci, že v současné době umí obě databáze reprezentovat objekty **výhradně jako body**. Každý objekt je lokalizován pomocí dvojice souřadnic na ploše. DBpedia sice používá pro popis souřadnic i vlastnost `geo:geometry`, která umožňuje definovat i složitější geometrické objekty, ale v databázi DBpedia jsou použity výhradně souřadnice zeměpisné délky a zeměpisné šířky, převážně jako desetinná čísla. Důvodem použití pouze bodů, je zdroj dat, tj. Wikipedie, ve které jsou objekty prostorově určeny vždy právě pomocí zeměpisných souřadnic. V databázi Wikidata je použit pro reprezentaci polohy speciální datový typ *zeměpisné souřadnice*, který je popsán níže. Ve vývojovém plánu je však zavedení nového datového typu, který by umožnil popsat přesněji geometrii objektu. Měl by se jmenovat *zeměpisný tvar* a vycházet z některého standardizovaného formátu, nejspíše GeoJSON nebo WKT.[1] Oba tyto formáty dokáží definovat základní geometrická primitiva, a jsou schopny i skládat prvky do složitějších geometrií, jako např. multipolygon. Formát WKT je koncipován i pro popis prostorových 2,5D a 3D objektů, jako např. povrch, TIN nebo mnohostěn.

- *Popis souřadnic* – Souřadnice jsou ve struktuře projektu Wikidata reprezentovány pomocí interního datového typu *zeměpisné souřadnice*, který je použit v celkem 8 datotypových vlastnostech. Kromě očekávaně nejpoužívanější vlastnosti „zeměpisné souřadnice“ (P625) je to ještě čtveřice vlastností „nejsevernější bod“ (P1332), „nejjižnější bod“ (P1333), „nejvýchodnější bod“ (P1334), „nejzápadnější bod“ (P1335), dále vlastnosti „souřadnice místa pohledu“ (P1259), „vztažný bod letiště“ (P2786), a jedna vlastnost určená k testování – „pískoviště – zeměpisné souřadnice“ (P626). Ačkoliv to není podrobně dokumentováno, z popisu položky ve formátu JSON lze vyčíst, že tento datový typ tvoří uspořádaná pětice 4 čísel a jednoho URI $\{latitude, longitude, altitude, precision, globe\}$, kde *latitude* reprezentuje „zeměpisnou“ šířku, *longitute* „zeměpisnou“ délku, *altitude* „nadmořskou“ výšku, *precision* podrobnost určení souřadnic a *globe* kosmické těleso, ke kterému se souřadnice vztahují. Uvedené pojmy jsou v uvozovkách z terminologických důvodů – Datový typ *zeměpisné souřadnice* se skutečně používá i pro popis objektů na povrchu Marsu, Titanu, nebo třeba Phobosu. Zde definované souřadnice lze jen těžko nazývat „zeměpisnými“, a označit výšku za

[1] https://phabricator.wikimedia.org/T57549

„nadmořskou“ je též zavádějící. V naprosté většině případů je však kosmickým tělesem „Země“ (Q2). Případů, kdy je hodnota odlišná, bylo zjištěno 6 344, z celkového počtu 2 631 281 položek, které obsahují vlastnost P625, tj. cca 0,2 % (u ostatních 7 vlastností se žádná výjimka nevyskytuje). Zvláštností je, že tento datový typ je navržen tak, aby umožňoval ukládat i informace o výšce, ale atribut *altitude* je označován za zastaralý[1] a není v současnosti využíván. Po exportu do RDF jsou souřadnice reprezentovány pomocí WKT.[2]

Ve znalostní bázi DBpedia je situace o něco více nepřehledná. Prostorově určené objekty jsou instancemi několika tříd, mezi nimiž je často definována dědičnost. Nejčastěji jsou prostorová data instancemi „nadtříd“ `geo:SpatialThing` a `yago:YagoPermanentlyLocatedEntity`[3]. Druhá zmíněná třída označuje objekty, které jsou převzaté z ontologie YAGO[4]. Vlastností, které slouží k polohovému určení objektů, je mnohem více, než v případě projektu Wikidata, přičemž jsou velmi často u objektů uváděny zároveň, v rozličných kombinacích. Nejčastěji se v databázi vyskytuje vlastnost `geo:geometry`, standardizovaná konsorciem W3C[5], která má vždy hodnotu datového typu `openlinks:geometry`, jenž má strukturu formátu WKT – `POINT(long lat)` – kde `long` a `lat` značí souřadnice zeměpisné délky a zeměpisné šířky v souřadnicovém systému WGS84, v tomto pořadí, převážně jako desetinná čísla. Druhou nejpopulárnější geolokalizační vlastností v databázi je `grs:point` (viz tabulka 4). Ta nejčastěji obsahuje datový typ `xsd:string`, formátovaný většinou jako dvojice čísel s desetinným rozvojem, oddělená mezerou, která reprezentují souřadnice zeměpisné délky a šířky (v tomto pořadí) v systému WGS84. Vzhledem k tomu, že extrakční algoritmy při tvorbě databáze nekontrolují správnost vkládaných hodnot, může se jako hodnota vlastnosti objevit objekt jakéhokoliv typu. Duplicitu používaných vlastností lze dokumentovat na výsledku vyhledávání v databázi, při kterém bylo zjištěno, že jen 3 prvky mají vlastnost `grs:point`, aniž by obsahovaly zároveň `geo:geometry`, přičemž tyto 3 objekty se nacházejí na jiném kosmickém tělese, než na Zemi, což je bezpochyby způsobeno chybnou exktrakcí z Wikipedie. Další obdobně rozšířená je dvojice standardizovaných vlastností `geo:lat` a `geo:long`, které určují šířku a délku v souřadnicích WGS84. Duplicitními vlastnosti k `geo:lat` a `geo:long` jsou vlastnosti `dbp:latitude` a `dbp:longitude`,

[1] https://www.mediawiki.org/wiki/Wikibase/DataModel/JSON#globecoordinate

[2] https://lists.wikimedia.org/pipermail/wikidata/2016-May/008648.html

[3] yago: <http://dbpedia.org/class/yago/>

[4] http://www.mpi-inf.mpg.de/departments/databases-and-information-systems/research/yago-naga/yago/

[5] http://www.w3.org/2003/01/geo/wgs84_pos

které jsou však využívány mnohem méně. Jinou vlastností je `dbo:location`[1]. Její hodnota má být instance třídy `dbo:Place`, což je i reálně dodržováno. Zcela nevhodně je pak používána snadno zaměnitelná vlastnost `dbp:location`. Její hodnota se liší od instancí třídy `dbo:Place`, přes dvě hodnoty `xsd:double` nebo `xsd:integer`, vyjadřující souřadnice s přesností odpovídající datovému typu, až po `xsd:string`, který má někdy tvar dvou čísel oddělených mezerou, někdy však jde jen o vágní slovní popis umístění objektu. Existují i případy, kdy je hodnotou jen jedno číslo, bez dalšího určení, tudíž není jasné, zda jde o zeměpisnou délku nebo šířku nebo něco úplně jiného, zejména v případech, kdy je číslo mimo interval $< -180, 180 >$. V případě, že je `dbp:location` použita zároveň s `dbo:location`, její hodnota je převážně odlišná. Dále pak existují vlastnosti, které, jsou-li použity, by se měly vyskytovat u dané položky najednou (dále jsou nazývány jako „skupinové vlastnosti“). Jde o skupinu `dbp:longew`, `dbp:latns`, `dbp:longd`, `dbp:latd`, `dbp:longm`, `dbp:latm`, `dbp:longs` a `dbp:lats`, které značí po řadě umístění na východní/západní polokouli (ve tvaru `E`/`W`), umístění na severní/jižní polokouli (ve tvaru `N`/`S`), celé stupně zeměpisné délky, celé stupně zeměpisné šířky, celé minuty délky, celé minuty šířky, vteřiny zeměpisné délky a vteřiny zeměpisné šířky. Poslední dvě jmenované se vyskytují jednak jako hodnoty datového typu `xsd:integer`, tak `xsd:double`. Zároveň však existují zcela duplicitní vlastnosti `dbp:longDirection`, `dbp:latDirection`, `dbp:longDegrees`, `dbp:latDegrees`, `dbp:longMinutes`, `dbp:latMinutes`, `dbp:longSeconds` a `dbp:latSeconds`, a dále též (rozdílné, neboť URI je „case sensitive“) `dbp:longEw`, `dbp:latNs`, `dbp:longD`, `dbp:latD`, `dbp:longM`, `dbp:latM`, `dbp:longS` a `dbp:latS`, stejného významu, jen s nižší četností výskytu v databázi. Jak je vidět v tabulce 4, tyto „skupinové vlastnosti“ mají podobnou četnost výskytu, ovšem existují i případy, kdy má objekt určenou zeměpisnou délku, ale ne šířku. Taktéž je z tabulky patrné, že vlastnosti, sloužící k určení polokoule výskytu, se používají v kombinaci s jinými, než s vlastnostmi „své“ skupiny. Poslední často použitou vlastností je `dbp:coordinates`, která je používána stejně rozličně jako `dbp:location`. Její hodnotou jsou nezřídkakdy i řetězce jako „see table below“ nebo „KPAE WPAE“, což zřejmě opět souvisí s kvalitou vyplnění patřičných Infoboxů na Wikipedii. V datasetu *DBpedia Wikidata* je používána převzatá vlastnost `wkdt:P625`[2]. Přehled hlavních tříd prostorových prvků a jejich četnost shrnuje tabulka 3, přehled nejvýznamnějších zmíněných vlastností, spolu s četností výskytu v databázi a nejčastěji použitými datovými typy, je uveden v tabulce 4.

[1] dbo: <http://dbpedia.org/ontology/>
[2] wkdt: <http://wikidata.dbpedia.org/resource/>

Tabulka 3: Třídy prostorových objektů v databázi DBpedia

Název třídy	*Počet výskytů*
`yago:YagoPermanentlyLocatedEntity`	1 257 964
`geo:SpatialThing`	948 247

Tabulka 4: Použité vlastnosti pro geolokalizaci prvků v databázi DBpedia

Název vlastnosti	*Nejčastější datový typ*	*Počet výskytů*
`geo:geometry`	`openlinks:geometry`	951 199
`grs:point`	`xsd:string`	948 247
`geo:lat`	`xsd:float`	951 199
`geo:long`	`xsd:float`	951 199
`dbp:longew`	`xsd:string`	344 405
`dbp:latns`	`xsd:string`	344 351
`dbp:longd`	`xsd:integer`	331 079
`dbp:latd`	`xsd:integer`	331 071
`dbp:location`	(různý)	299 808
`dbo:location`	`dbo:Place`	229 113
`dbp:longitude`	`xsd:double`	84 228
`dbp:latitude`	`xsd:double`	84 217
`dbp:latDirection`	`xsd:string`	47 260
`dbp:longDirection`	`xsd:string`	47 254
`dbp:longDegrees`	`xsd:integer`	46 944
`dbp:latDegrees`	`xsd:integer`	46 943
`dbp:longD`	`xsd:integer`	36 326
`dbp:latD`	`xsd:integer`	36 326
`dbp:longEw`	`xsd:string`	32 487
`dbp:latNs`	`xsd:string`	32 485
`dbp:coordinates`	(různý)	1 523

- *Podpora různých souřadnicových systémů* – Vzhledem k tomu, že v databázi Wikidata je zacházeno se všemi souřadnicemi jako se zeměpisnými, ukládání v různých souřadnicových systémech, natož pak jejich vzájemná konverze, podporováno není. Zároveň však projekt Wikidata však umožňuje popsat polohu objektu ve sférických souřadnicích i na jiném tělese, než je Země, je-li toto těleso samo reprezentováno položkou v databázi Wikidata. V databázi DBpedia žádná z mnoha použitých vlastností není určena k polohové lokalizaci prvků v jiném souřadnicovém systému, než WGS84.

- *Podrobnost souřadnic* – V projektu Wikidata slouží k vyjádření podrobnosti souřadnic prvek `geoPrecision` v datovém typu *zeměpisné souřadnice*. Vyplněný datový typ *zeměpisné souřadnice* včetně přesnosti po exportu do RDF je na ukázce 1. V databázi DBpedia lze posoudit podrobnost souřadnic podle použi-

tých vlastností pro jejich popis. Je-li u objektu uvedena např. vlastnost `dbp:latd` s celočíselnou hodnotou, ale není již uvedena vlastnost `dbp:latm` ani `dbp:lats`, je potřeba pracovat s nepřesností určení zeměpisné šířky v řádu desetin stupně.

Ukázka 1: RDF reprezentace datového typu *zeměpisné souřadnice* s vyplněnou přesností v databázi Wikidata pro položku „Fakulta aplikovaných věd“ (Q11988196) ve formátu Turtle.

```
1  @prefix wikibase: <http://wikiba.se/ontology#> .
2  @prefix wd: <http://www.wikidata.org/entity/> .
3  @prefix wds: <http://www.wikidata.org/entity/statement/> .
4  @prefix wdv: <http://www.wikidata.org/value/> .
5  @prefix p: <http://www.wikidata.org/prop/> .
6  @prefix psv: <http://www.wikidata.org/prop/statement/value/> .
7  wd:Q11988196 p:P625 wds:Q11988196-6D75ADB9-DD67-4565-9F76-820
       B79F46454 .
8  wds:Q11988196-6D75ADB9-DD67-4565-9F76-820B79F46454 a wikibase:
       Statement,
9            psv:P625 wdv:4a0892d265b8b22d261f8586e1c317db ;
10 wdv:4a0892d265b8b22d261f8586e1c317db a wikibase:
       GlobecoordinateValue ;
11           wikibase:geoLatitude "49.7247"^^xsd:decimal ;
12           wikibase:geoLongitude "13.3503"^^xsd:decimal ;
13           wikibase:geoPrecision "0.01"^^xsd:decimal ;
14           wikibase:geoGlobe <http://www.wikidata.org/entity/Q2> .
```

3.3.2 Prostorové vztahy

- *Vyjádření prostorových vztahů* – Prostorové vztahy mezi prvky mohou být ve znalostních bázích, založených na grafové struktuře popsány pomocí za tímto účelem definovaných vlastností. Poněkud překvapivě zatím takové vlastnosti nejsou pevně standardizovány. V každé z řešených databází je snaha popsat prostorové vztahy alespoň částečně. Ne tolik překvapivě jsou k tomuto účelu použity v každé znalostní bázi vlastnosti jiné. Obecně se ale přístup obou projektů shoduje v tom, které prostorové vztahy mezi objekty se snaží modelovat, a které ne.

- *Vztah ekvivalence* – V obou databázích existují pouze vlastnosti pro popis sémantické ekvivalence. Prostorová totožnost objektů není nijak řešena. Prostorovou ekvivalenci je možné pouze odvodit ze shodné dvojice souřadnic.

- *Disjunktní vztah* – Ani pro vyjádření disjunkce neexistuje v řešených databázích žádný explicitní popis. V databázi Wikidata sice existuje vlastnost „rozdílné od“ (P1889), ale opět nejde o vyjádření prostorového vztahu mezi položkami, ale obvykle o rozlišení prvků se stejným názvem.

- *Vztah sousedství* – V projektu Wikidata je vztah sousedství jedním z mála dobře vyjádřených prostorových vztahů mezi objekty. Existuje symetrická vlastnost „hraničí s“ (P47), která označuje prvky se sdílenou hranicí. Je používána především u objektů administrativního členění – států a správních jednotek odpovídající si úrovně. V databázi DBpedia existují dvě velmi podobné vlastnosti, a sice `dbo:neighboringMunicipality`, která je použita u 2 761 objektů a `dbp:neighboringMunicipalities`, použitá u 2 762 objektů. Přitom vždy, až na jeden případ, jsou uvedeny obě vlastnosti zároveň.

- *Vztahy „obsahuje“ a „je obsažen v“* – Projekt Wikidata má pro vyjádření polohy prvku uvnitř jiného objektu zavedenu především vlastnost „nachází se v administrativní jednotce“ (P131). Krom toho existuje též širší vlastnost „stát“ (P17). Pomocí těchto vlastností lze snadno nalézt prvky, které se nacházejí ve vymezených administrativních hranicích, i naopak hledat objekty uvnitř těchto hranic. Ve znalostní bázi DBpedia je pro popis inkluze objektů použito více vlastností. Umístění prvku uvnitř většího celku může být popsáno například pomocí široce použitelných vlastností `dbo:location` a `dbp:location`.

- *Vztah „overlay“* – Vzájemný vztah objektů s neprázdným průnikem. Není ani v databázi Wikidata ani v databázi DBpedia modelován, ale reasoner může takový vztah odvodit z vícenásobného použití vlastností uvedených v předchozím odstavci. Je-li např. vlastnost „nachází se v administrativní jednotce“ (P131) v datové bázi Wikidata uvedena s více hodnotami, lze odvodit, že položka, u které je vlastnost uvedena, leží alespoň svou částí ve všech uvedených jednotkách, a je-li její oblast spojitá, pak musí tyto jednotky spolu sousedit, a položka musí mít též neprázdný průnik s jejich společnou hranicí, nebo ji zcela překrývat. Zda se jedná o *overlay*, překrytí nebo obsažení, pak závisí na konkrétní situaci. Obdobně to platí i pro projekt DBpedia.

- *Vztahy „překrývá“ a „je překryt“* – Ani v jedné z databází v současné době nejsou implementovány vlastnosti, které by popisovaly vztah objektů ve více vrstvách. Tj. ani vztah vzájemného překrývání prvků není vyjádřen.

Tabulka 5: Porovnání prostorových vlastností dat

Parametr	**DBpedia**	**Wikidata**
Geometrie	Jen body	Jen body
Popis souřadnic	Standardizovaný, nejednotný	Nestandardní, jednotný
Podpora různých souřadnicových systémů	Ne	Ne
Podrobnost souřadnic	Neuvedena	Rozdílná
Vztah ekvivalence	Ne	Ne
Disjunktní vztah	Ne	Ne
Vztah sousedství	`dbp:neighboringMunicipalities` `dbo:neighboringMunicipality`	`wd:P47`
Vztah *overlay*	Ne	Ne
Vztahy „obsahuje“ a „je obsažen v“	`dbo:location` `dbp:location`	`wd:P131` `wd:P17`
Vztahy „překrývá“ a „je překryt“	Ne	Ne

3.4 Kvantitativní porovnání

Z hlediska využitelnosti dat je velice důležitá úplnost popisu reálného světa. Tu lze alespoň přibližně odhadnout podle některých číslech, které v daném čase databázi popisují. *a*) *Počet podporovaných jazyků* je důležitý z hlediska globální využitelnosti dat. V čím větším množství jazyků databáze obsahuje popisy objektů, tím pro větší množství uživatelů je možné nad daty postavenou aplikaci přizpůsobit. V tomto směru je projekt Wikidata o notný kus napřed před databází DBpedia. *b*) *Počet konceptů* zhruba vyjadřuje, jak podrobně daná znalostní báze popisuje svou doménu, tj. jak kvalitně plní svou funkci. *c*) *Počet RDF trojic* je druhým měřítkem, kterým lze poměřit kvalitu popisu domény. Zjednodušeně přibližně platí, že čím více trojic, tím dokonalejší popis objektů. To ale může být značně ovlivněno např. duplicitními vlastnostmi. *d*) *Počet RDF trojic na jeden koncept* dává představu, kolik informací je v databázi dostupných o každém jejím prvku. *e*) *Počet konceptů se souřadnicemi* je specifickým ukazatelem pro oblast prostorových dat. Přibližně ukazuje, jak velká část databáze je použitelná pro kartografickou vizualizaci.

Tabulka 6: Porovnání aktuálního počtu dat v databázích

Parametr	**DBpedia**	**Wikidata**
Počet podporovaných jazyků	127	358
Počet popisovaných konceptů	16 947 148	17 518 028
Počet RDF trojic	883 644 351	57 089 562
Počet trojic na 1 koncept	50,2	4,09
Počet konceptů se souřadnicemi	cca 1 130 000 (cca 6,7 %)	2 629 890 (15 %)

4 Využitelnost prostorových dat z projektů DBpedia a Wikidata

4.1 Omezení možností vizualizace prostorových dat

Z porovnání provedeného v kapitole 3 vyplývají některá omezení pro kartografickou vizualizaci. Jak již bylo zmíněno v sekci 3.3.1, nejzásadnějším faktem je, že v obou řešených databázích jsou (alespoň prozatím) objekty geometricky popsány jen jako body ve dvoudimenzionálním prostoru. To znamená, že objekty, které jsou v databázích popisované a mají složitější geometrii, jako jsou státy, administrativní územní celky, vodní plochy, města, geografické oblasti, silnice, železnice, vodní toky, budovy, geomorfologické celky a mnohé další, je většinou nevhodné vizualizovat pomocí souřadnic u nich uvedených. Nejjednodušší možností, jak se s tímto problémem vypořádat, je využít odkazů do jiných projektů, které používají pro popis objektů složitější geometrii, zejména pak OpenStreetMap. Wikidata jsou propojena s OpenStreetMap pomocí vlastností, popsaných již v sekci 1.5. DBpedia je pak propojena s ontologií LinkedGeoData, která je RDF reprezentací projektu OpenStreetMap.

Druhým omezením, na které je potřeba dbát, je prostorová povaha souřadnic objektů. Všechny objekty, v obou databázích, jsou-li lokalizovány pomocí souřadnic, pak jsou v geocentrickém systému WGS84, tj. na referenčním elipsoidu. Pro zobrazení takto určených objektů v rovině je potřeba nejdříve souřadnice transformovat pomocí vhodně zvoleného kartografického zobrazení.

Pro využití prostorových dat ve znalostní bázi DBpedia platí jasné omezení, kterým je nejednotnost popisu souřadnic.

Nevýhodou je, že ani v jednom z projektů nejsou souřadnice určovány i s výškou. Nelze tak použít data pro 3D vizualizace. V databázi DBpedia je sice použita vlastnost `dbo:elevation`, ale je uvedena zejména u významných vrcholů (celkem 184 515 použití). Obdobně v databázi Wikidata existuje vlastnost „nadmořská výška“ (P2044), která je použita u 296 840 položek.

4.2 Známé chyby v prostorových datech

4.2.1 Vytěžení dat

Před samotnou analýzou prostorových dat v obou řešených databázích, bylo nejprve potřeba správná data vybrat a uložit ve formátu, který umožní jednoduché zpracování. Pro tento účel byla, jako vedlejší produkt celé práce, vytvořena javascriptová aplikace

LOD Ion Cannon, která se vypořádává se dvěma zásadními problémy:

1. Vybrat potřebná data z RDF databáze je pomocí jazyka SPARQL jednoduché. Veřejné SPARQL endpointy mají však nastaveno omezení na maximální počet vrácených výsledků, aby snížily zatížení serverů od přenosů velkých souborů dat. Konkrétně pro SPARQL endpoint databáze DBpedia to v roce 2016 činilo 10 000 záznamů. Takový počet výsledků je vzhledem k počtům položek se souřadnicemi, uvedeným v sekci 3.4, velmi neuspokojivý. Adekvátní SPARQL dotaz je proto potřeba mnohokrát opakovat s patřičným posunutím výsledků dotazu (pomocí klíčového slova `OFFSET`). *LOD Ion Cannon* tento proces zjednoduší tak, že pro zadaný dotaz nejprve zjistí celkový počet výsledků dotazu (pomocí agregační funkce `Count()`), a poté odešle na server potřebný počet parciálních dotazů, lišících se jen vzájemným posunutím seznamu výsledků. Tato operace je na pozadí prováděna přes asynchronní AJAX žádost, za použití rozhraní XMLHttpRequest. Celé řešení je tedy založeno na nejjednodušším možném a nejméně efektivním způsobu obejití nastaveného limitu, a odtud také plyne název „Ion Cannon“.

2. Druhým problémem je takto fragmentovaná data opět spojit a uložit do snadno vizualizovatelného formátu. Jako cílový formát byl zvolen textový soubor CSV, který lze snadno otevřít např. v programu QGIS[1]. *LOD Ion Cannon* tedy nejprve převede výsledek parciálního dotazu z formátu JSON do CSV a jako BLOB jej uloží do RAM. Po stažení a převedení všech výsledků jsou fragmenty spojeny dohromady a nabídnuty uživateli k uložení na disk. Takováto práce se soubory je v JavaScriptu možná díky rozhraní File API, které ale bylo v dubnu 2016 pouze ve stádiu „Working Draft“, a proto byla část chování simulována knihovnou FileSaver[2]. Konečným výstupem je tedy jediný CSV soubor, který by byl i výsledkem původního, nerozděleného dotazu. Část výsledného CSV souboru pro DBpedia a vlastnost `geo:geometry` je na ukázce 2.

Ukázka 2: Část CSV souboru, obsahujícího data získaná SPARQL dotazem

```
1 place,wkt
2 "http://dbpedia.org/resource/Rosendal,_Free_State","POINT
     (27.916666030884 28.5)"
3 "http://dbpedia.org/resource/Samiopoula","POINT(26.79400062561
     37.627998352051)"
4 "http://dbpedia.org/resource/Hastijan","POINT(50.776390075684
     33.860553741455)"
```

[1] http://www.qgis.org/
[2] https://github.com/eligrey/FileSaver.js

```
5 "http://dbpedia.org/resource/Reception_Tower_Utlandshörn","POINT
    (7.10777759552 53.563056945801)"
```

Při konstrukci SPARQL dotazů bylo potřeba pečlivě dbát na přesnou specifikaci jazyka SPARQL, zejména pro klíčové slovo DISTINCT. Na rozdíl od jazyka SQL, ve SPARQL nelze tento příkaz aplikovat na jednotlivé proměnné, ale jen na množinu proměnných jako celek. Dotaz s hlavičkou `SELECT DISTINCT ?s ?p ?o` tak nehledá jednoznačné výsledky pouze pro proměnnou `?s`, ale pro trojici `?s ?p ?o`.[1] V konkrétním případě hledání položek se souřadnicemi to znamená, že v případě, kdy je u některé položky uvedeno více souřadnic (typicky u objektů s velkou plochou), je výsledkem dotazu tato položka tolikrát, kolik souřadnic, resp. kombinací souřadnic obsahuje. Vzhledem k tomu, že všechny uvedené souřadnice se většinou zásadně neliší, bylo toto chování vyřešeno tak, že z uvedených souřadnic je vybrána pouze jedna náhodná, pomocí funkce `SAMPLE()`. Použitý SPARQL dotaz pro získání položek z databáze DBpedia s vlastností `geo:geometry` je na ukázce 3, `OFFSET` se navyšuje o 10 000 s každou iterací programu. Kompletní SPARQL dotazy jsou shrnuty v příloze A.

Ukázka 3: SPARQL dotaz pro získání dat se souřadnicemi

```
1 SELECT DISTINCT ?place SAMPLE(?wkt) AS ?wkt WHERE {
    ?place geo:geometry ?wkt.
3 } GROUP BY ?place
  LIMIT 10000 OFFSET 10000
```

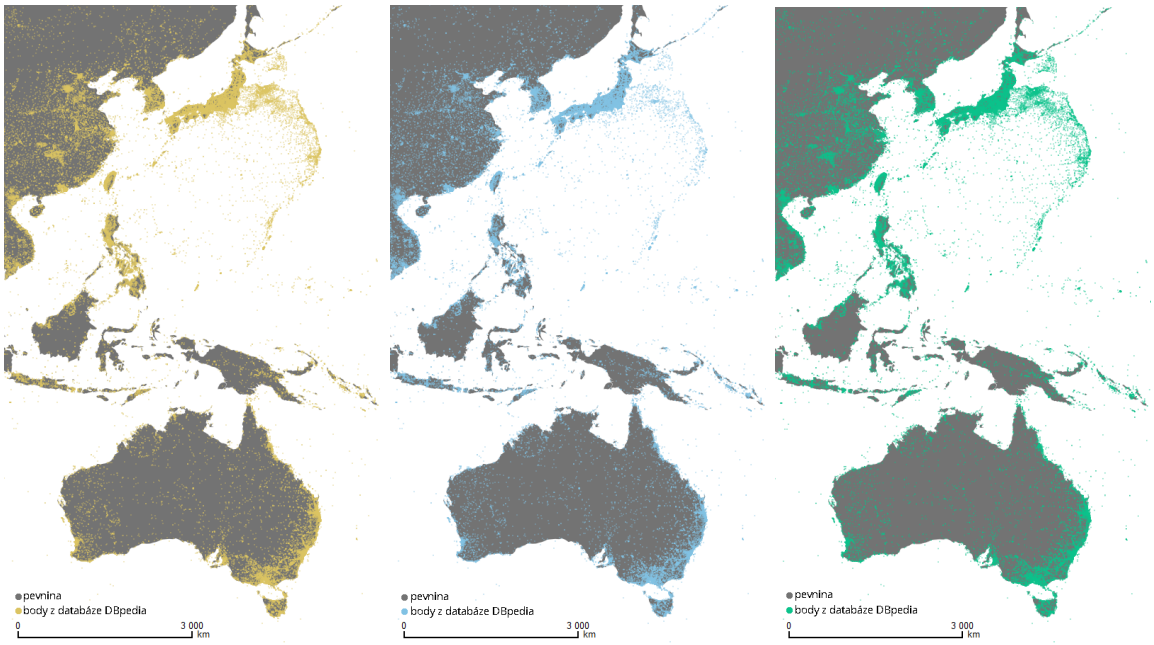
Tímto způsobem tedy byly z databáze Wikidata staženy všechny položky, které mají vlastnost P625 („zeměpisné souřadnice“), a všechny položky databáze DBpedia, které mají alespoň jednu z vlastností či kombinací vlastností `geo:geometry`, `grs:point`, (`geo:lat` ∧ `geo:long`), ((`dbp:latd` ∧ `dbp:longd`) ∨ (`dbp:latD` ∧ `dbp:longD`)). Zdrojový kód aplikace *LOD Ion Cannon* je dostupný pod licencí MPLv2 na serveru GitHub[2].

4.2.2 Analýza prostorových dat v databázi DBpedia

V roce 2012 popsal Jordi Castells Sala v [26] dvě chyby v prostorových datech databáze DBpedia. První z nich je tzv. zrcadlová Austrálie – efekt, který vznikl špatnou extrakcí dat z Wikipedie, kde pravděpodobně došlo k nechtěnému zdvojení souřadnice zeměpisné šířky, jednou se záporným znaménkem a jednou bez něj. To se při vizualizaci projeví tak, že položky, které se mají nacházet pouze na jižní polokouli se zároveň zobrazí na odpovídající poloze severní zeměpisné šířky. Jelikož se tato chyba týká takřka

[1] http://stackoverflow.com/a/5397931
[2] https://github.com/jmacura/LOD-Ion-Cannon

výhradně dat z oblasti Austrálie, dojde k zobrazení jakési převrácené Austrálie na severní polokouli. Ačkoliv v [26] je tato chyba popisována jen na vlastnosti `geo:lat` (viz obrázek 7a), týká se v obdobné míře i vlastností `grs:point` a `geo:geometry`, jak je vidět na obrázku 7b, resp. 7c.

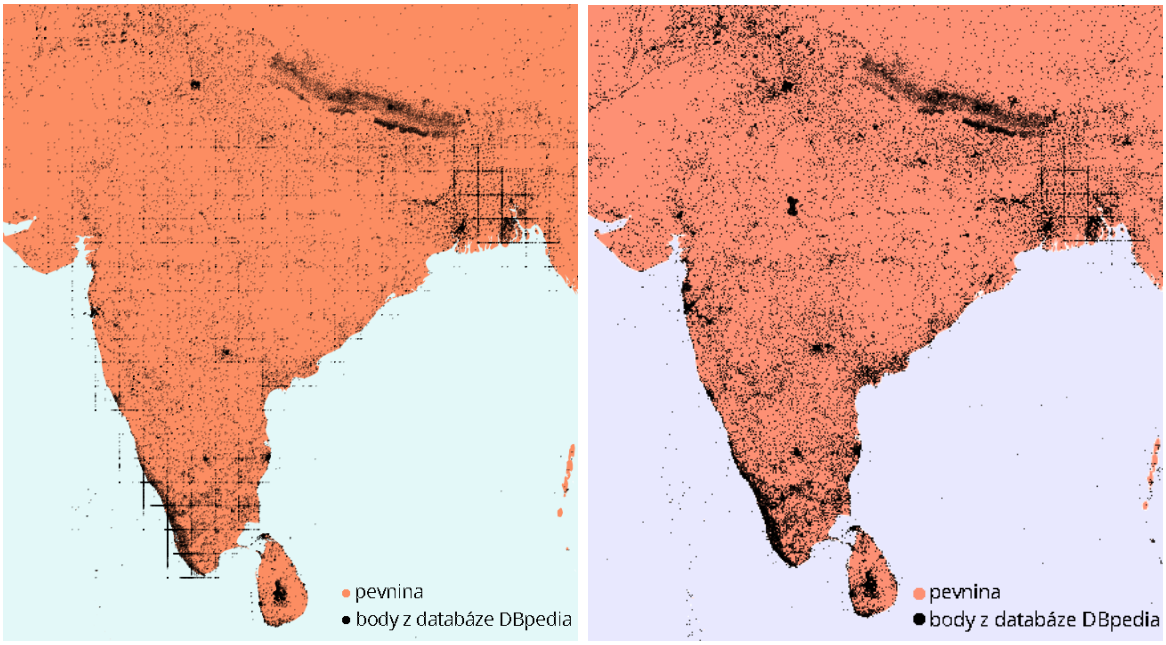


(a) Vlastnosti `geo:lat` a `geo:long`. (b) Vlastnost `grs:point`. (c) Vlastnost `geo:geometry`.

Obrázek 7: Zobrazení zrcadlové Austrálie u bodů se souřadnicemi z databáze DBpedia, podle ovlivněných vlastností.

Druhou chybou, dokumentovanou v [26], je mřížka, která se zobrazuje nad celou oblastí Indie a Bangladéše, jak je vidět na obrázku 8a. *Mřížka* vzniká tak, že velké množství bodů leží na rovnoběžkách a/nebo polednících. Jordi Castells Sala to přičítá zaokrouhlovací chybě. Tato mřížka ale vůbec nevzniká, pokud jsou data získána SPARQL dotazem s konstrukcemi `GROUP BY` a `Count()`. Při prozkoumání vlastností jednotlivých položek vyjde najevo, že většina z těchto položek, které se zobrazují v mřížce, má uvedeno několik různých souřadnic, a body na mřížce jsou výsledkem kartézského součinu. V současnosti se zobrazení mřížky podařilo simulovat jen v oblasti Bangladéše, v prostoru Indie již bylo uvádění více souřadnic eliminováno. Konkrétně bylo nalezeno u vlastností `geo:lat` a `geo:long`, a též vlastnosti `geo:geometry`. Kartografická vizualizace těchto tří vlastností v problematické oblasti je na obrázku 8b.

Určitým omezením je též definice vlastností `dbp:latDegrees` a `dbp:longDegrees`, resp. `dbp:latd` a `dbp:longd`, včetně `dbp:latD` a `dbp:longD`, konkrétně jejich rozsah. Tím totiž mají být pouze kladné hodnoty (typu `xsd:integer`). Poloha bodu v rámci konkrétního zemského kvadrantu je uvedena zvlášť ve vlastnostech `dbp:latDirection`



(a) Mřížka nad Indií a Bangladéší v roce 2012. (Převzato z [26])

(b) Mřížka nad Bangladéší v roce 2016.

Obrázek 8: Souřadnice v mřížce, jakožto výsledek kartézského součinu více hodnot souřadnic, v Indii a Bangladéši.

a `dbp:longDirection`, resp. `dbp:latns` a `dbp:longew`, včetně `dbp:latNs` a `dbp:longEw`, jejichž rozsahem by měly být pouze hodnoty typu `xsd:string`.

4.2.3 Vybraná specifika prostorových dat v databázi Wikidata

Hlavním úskalím dat z projektu Wikidata je fakt, že souřadnice obsahují i položky, které se nemusí nutně nacházet na Zemi. Při každém dotazu, jehož výsledkem jsou geografické objekty, je potřeba zajistit, aby datotypová vlastnost `wikibase:geoGlobe` měla hodnotu Q2 – „Země“. Anebo toho lze samozřejmě využít. Např. dotaz na ukázce 4 zobrazí seznam nejvyšších hor ve sluneční soustavě; přesněji řečeno, *nejvyšších hor ze všech hor známých pro projekt Wikidata*. Část výsledku dotazu je na obrázku 9.

Ukázka 4: SPARQL dotaz na nejvyšší hory v databázi Wikidata

```
SELECT ?subj ?label ?coord ?elev
WHERE
{
  ?subj wdt:P2044 ?elev
  FILTER(?elev > 8000) .
  ?subj wdt:P625 ?coord .
  ?subj rdfs:label ?label .
```

```
8 } ORDER BY ?elev
```

Stejně tak Wikidata obsahují souřadnice i u položek, u kterých to není příliš očekávané. Především jde o jednotlivé rovnoběžky a poledníky, časová pásma, aj. Tyto objekty je většinou potřeba z výsledků prostorových dotazů odfiltrovat.

subj	label	coord	elev
wd:Q520	Olympus Mons	<http://www.wikidata.org/entity/Q111> Point(-133.8 18.65)	21229
wd:Q499164	Ascraeus Mons	<http://www.wikidata.org/entity/Q111> Point(-104.08 11.92)	18225
wd:Q18002964	Boösaule Montes	<http://www.wikidata.org/entity/Q3123> Point(-90.89 -3.75)	17850
wd:Q704936	Arsia Mons	<http://www.wikidata.org/entity/Q111> Point(-120.09 -8.35)	17761
wd:Q735657	Pavonis Mons	<http://www.wikidata.org/entity/Q111> Point(-112.85156 0.58227)	14058
wd:Q924119	Elysium Mons	<http://www.wikidata.org/entity/Q111> Point(147.21 25.02)	14028
wd:Q1027482	Caltech Submillimeter Observatory	Point(-155.476 19.8225)	13570
wd:Q21838032	Ionian Mons	<http://www.wikidata.org/entity/Q3123> Point(236.24 9.04)	12700
wd:Q21839566	Hi'iaka Montes	<http://www.wikidata.org/entity/Q3123> Point(81.96 -4.68)	11100
wd:Q21839565	Haemus Montes	<http://www.wikidata.org/entity/Q3123> Point(46.6 -70.12)	10800
wd:Q18356771	Selenean summit	<http://www.wikidata.org/entity/Q405> Point(201.3665 5.4125)	10786
wd:Q3486102	Skadi Mons	<http://www.wikidata.org/entity/Q313> Point(4.0 64.0)	10700
wd:Q21839553	Caucasus Mons	<http://www.wikidata.org/entity/Q3123> Point(238.48 -31.95)	10600
wd:Q16509686	Euboea Montes	<http://www.wikidata.org/entity/Q3123> Point(335.78 -47.94)	10500
wd:Q21839563	Gish Bar Mons	<http://www.wikidata.org/entity/Q3123> Point(88.95 18.5)	10350
wd:Q3045546	Titan Dome	Point(165.0 -88.5)	10170
wd:Q5348327	Egypt Mons	<http://www.wikidata.org/entity/Q3123> Point(257.6 -41.49)	10000
wd:Q21839552	Capaneus Mensa	<http://www.wikidata.org/entity/Q3123> Point(121.4 -16.82)	9350
wd:Q21839555	Dorian Montes	<http://www.wikidata.org/entity/Q3123> Point(196.69 -25.1)	8850
wd:Q513	Mount Everest	Point(86.925277777 27.988055555)	8848
wd:Q3375694	Tohil Mons	<http://www.wikidata.org/entity/Q3123> Point(161.57 -28.42)	8800
wd:Q21298740	South Summit	Point(86.925833333 27.985)	8749
wd:Q43512	K2	Point(76.513333333 35.881111111)	8611

Obrázek 9: Výsledek dotazu „Nejvyšší hory ve sluneční soustavě“ z databáze Wikidata. Ve sloupečku *coord* lze vidět na jakém tělese daná hora leží – např. položka Q111 je Mars. U *Caltech Submillimeter Observatory* a *Titan Dome* je výška uvedena ve stopách.

Na obrázku 9 lze zároveň vidět, že podle výsledků dotazu, Mount Everest není nejvyšší horou na Zemi. Tato chybná interpretace je způsobena neschopností SPARQL endpointu automaticky převádět jednotky, v tomto případě stopy na metry. V ukázce je totiž použit přístup z položky přímo k hodnotě vlastnosti pomocí vlastnosti ze jmenného prostoru `wdt:`[1]. Tento přístup ignoruje jednotky uvedené v datovém typu *množství* a nabízí pouze hodnotu zadanou editorem projektu Wikidata. Pro zjištění jednotek by bylo potřeba nejprve přistoupit k celému tvrzení pomocí vlastnosti `p:P2044`[2], poté k uzlu datového typu *množství* pomocí vlastnosti `psv:P2044`[3] a konečně k použité jed-

[1] wdt: <http://www.wikidata.org/prop/direct/>
[2] p: <http://www.wikidata.org/prop/>
[3] psv: <http://www.wikidata.org/prop/statement/value/>

notce pomocí vlastnosti `wikibase:quantityUnit`. Rozšířený SPARQL dotaz o uvedení jednotek je v příloze A.

4.3 Existující aplikace využívající prostorová data z projektů Wikidata nebo DBpedia

Jednou z prvních aplikací, které využívají prostorových dat z projektu DBpedia, byla aplikace DBpedia Mobile. Ta je určena pro mobilní zařízení a umožňuje uživateli zobrazit položky, které se nacházejí v jeho okolí, a navíc následuje odkazy z projektu DBpedia do dalších LOD datasetů, tudíž nabízí uživateli také informace z těchto propojených datasetů. Toto propojení LOD databází v jedné člověkem čitelné stránce je univerzální řešení, které se jmenuje *Marbles Linked Data Engine*.

Modernějším nástupcem DBpedia Mobile by mohla být aplikace pro OS Android, jménem DBpedia Places[1]. Obdobně jako DBpedia Mobile umí vyhledat položky v databázi DBpedia, které se nacházejí v okolí uživatele, ale navíc je zobrazí nad mapou Google maps a jednotlivá místa opatří odkazem na odpovídající článek ve Wikipedii.

Rozhraním, které může sloužit pro procházení geolokalizovaných položek databáze DBpedia, je projekt OOBIAN[2]. Ten vedle zobrazení položek v člověkem čitelné formě, či procházení stromu jednotlivých konceptů, nabízí také velmi zajímavé procházení pomocí kategorií (tzv. „OOBIAN Drill“), a především prohlížení položek v mapovém okně (tzv. „OOBIAN Maps“). Tato aplikace má tu výhodu, že uživatel snadno vidí všechny geolokalizované položky ve vybraném zájmovém území.

Aplikací se značným, ale doposud nevyužitým potenciálem, je projekt Vagueplaces Generator, který vznikl v roce 2012 na Univerzitě v Twente. Tento projekt byl zaměřen na definování hranic neurčitých míst, tj. oblastí, které nekopírují administrativní hranice, jako je např. oblast The Midlands v Anglii, Helgeland v Norsku, či Chodsko v Česku. Projektů, které by se snažily dosáhnout obdobného cíle existuje více, ale Vagueplaces Generator se odlišuje využitím databáze DBpedia jako zdroje vstupních dat. Výsledkem projektu je program napsaný v jazyce Python, který pro zadané klíčové slovo či slova (např. „The Midlands“) nejprve nalezne položky z databáze DBpedia, které se nacházejí v hledané oblasti a poté zpracuje jejich souřadnice tak, aby byla uživateli vrácena nejpřesnější možná hranice oblasti ve formátu WKT. [26] Zdrojový kód projektu je k nalezení na serveru GitHub.com[3].

[1] https://play.google.com/store/apps/details?id=com.lauer.dbpediaplacesandroid
[2] http://dbpedia.oobian.com
[3] https://github.com/kxtells/vague-places

Příkladem projektu, který využívá dat uložených v databázi Wikidata a má potenciál využití v kartografických aplikacích je qLabel[1]. Jde o javascriptovovou knihovnu, určenou pro jazykovou lokalizaci názvů geoprostorových prvků. Principem je, že pro každý např. místopisný název, v libovolném jazyce, je možné v databázi Wikidata dohledat *štítek* odpovídající položky, a to v jazyce uživatele. Knihovna je distribuována jako open source a jejím vývojářem je firma Google.

Ukázkovým využitím prostorových dat v obou projektech je mapa „geografie násilí“, vytvořená holandskou firmou LAB1100. Tato mapa zobrazuje všechny bitvy, které se v dějinách lidstva odehrály pomocí teček v místech jejich bojiště. Navíc tečky odlišuje metodou heat-map podle období, ve kterém se bitvy odehrály. Výsledná mapa je dostupná na webu[2]. [28]

4.4 Vytvoření vlastní vizualizace dat

Tato část popisuje způsob vytvoření ukázkové webové mapové aplikace, která využívá data z projektů DBpedia a Wikidata. Cílem je ukázat, jak může být s daty pracováno, jakým způsobem mohou být získána a jak je lze vizualizovat ve webovém prostředí. Řešení je vytvořeno v programovacím jazyce JavaScript, s využitím knihoven Leaflet[3], která umožňuje snadnou manipulaci s prostorovými daty, a jQuery[4], která usnadňuje práci s AJAXem.

Knihovna Leaflet umožňuje jednoduchým způsobem vytvořit dynamické mapové okno uvnitř HTML stránky. Objekt mapového okna je v DOM stránky reprezentován elementem `DIV` s atributem `id="map"`. Další manipulace s tímto objektem již probíhá jen pomocí JavaScriptu nebo CSS. Po vytvoření objektu mapového okna je potřeba přidat do něj podkladovou mapu. Ta se vytvoří funkcí `L.tileLayer()` s parametrem, kterým je URI pro stažení mapových dlaždic. Nejčastěji se pro tento účel využívá projektu Mapbox[5], který poskytuje dlaždice renderované z OpenStreetMap na základě přednastavených nebo uživatelem definovaných stylů. Tento základní postup provede kód ukázky 5.

[1] http://googleknowledge.github.io/qlabel/
[2] http://battles.nodegoat.net/viewer.p/23/385/scenario/3/geo/fullscreen
[3] http://leafletjs.com/
[4] http://api.jquery.com/
[5] https://www.mapbox.com/

Ukázka 5: Vytvoření mapového okna a přidání podkladové vrstvy pomocí knihovny Leaflet

```
var map = L.map('map').setView([49.77, 13.38], 12);
L.tileLayer('
  http://{s}.tiles.mapbox.com/v3/yjm.j5j87g72/{z}/{x}/{y}.png',
  {maxZoom: 18}
).addTo(map);
```

4.4.1 Dotaz přes WikidataQuery API

WikidataQuery API je standardní RESTful rozhraní pro přístup k datům v databázi Wikidata. Základní adresa tohoto API je `https://wdq.wmflabs.org/api?q=`. Použitou funkcí je `AROUND[lat, lon, dist]`, která vrátí seznam položek ve vzdálenosti `dist` od souřadnic `lat, lon`. Výsledkem dotazu je JSON, který obsahuje všechny informace odstupné o položkách. Pro zmenšení objemu přenášených dat je vhodné toto omezit na vybrané vlastnosti pomocí parametru `props=`, např. jen na vlastnost P625, tedy zeměpisné souřadnice. V tomto příkladě jsou podle výsledků ve formátu JSON vytvořeny značky na získaných souřadnicích, a ty jsou poté přidány do nové vrstvy v mapovém okně. Tento postup je shrnut v ukázce 6.

Ukázka 6: Získání a zpracování dat z WikidataQuery API

```
$.getJSON('https://wdq.wmflabs.org/api?q=around%5B625
    ,49.7462414,13.3778353,2%5D&props=625' + '&callback=?', function
    (data) {
  var d = data.props[625];
  var wd_layer = L.featureGroup(null);
  for(var i = 0; i < d.length; i++) {
    var coords = d[i][2].split("|");
    var mark = L.marker([coords[0],coords[1]]);
    wd_layer.addLayer(mark);
  }
  wd_layer.addTo(map);
  controller.addOverlay(wd_layer, "Items around given coordinates")
      ;
});
```

4.4.2 Dotaz přes WikidataQuery Service SPARQL endpoint

Druhou a mnohem silnější možností pro přístup k datům projektu Wikidata je dotazování přímo přes SPARQL endpoint. Práce s ním je velmi obdobná jako v případě předchozím. Klíčové je správně sestavit SPARQL dotaz, kterým chceme data získat. V tomto případě jde o ukázkový příklad „Najdi města a jejich souřadnice, která mají v čele starostku (nebo primátorku)“. SPARQL dotaz je formulován v ukázce 7. Tento dotaz je možné rozšířit – např. místo ID pro město a jeho představitelku můžeme chtít získat jejich jména, pokud jsou vyplněna, a odkazy na články v anglické Wikipedii, které se jich týkají, pokud existují. Přesně tyto volitelné části jsou zobrazeny v ukázce 8.

Ukázka 7: SPARQL dotaz pro nalezení měst, která mají v čele ženu, z databáze Wikidata

```
PREFIX wikibase: <http://wikiba.se/ontology#>
PREFIX : <http://www.wikidata.org/entity/>
PREFIX wdt: <http://www.wikidata.org/prop/direct/>
PREFIX p: <http://www.wikidata.org/prop/>
PREFIX ps: <http://www.wikidata.org/prop/statement/>
PREFIX psv: <http://www.wikidata.org/prop/statement/value/>
PREFIX pq: <http://www.wikidata.org/prop/qualifier/>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
SELECT DISTINCT ?lat ?lon ?citylabel ?cityarticle ?mayorlabel ?
    mayorarticle WHERE {
  ?city wdt:P31/wdt:P279* :Q515 .
  ?city p:P6 ?statement .
  ?statement ps:P6 ?mayor .
  ?mayor wdt:P21 :Q6581072 .
  FILTER NOT EXISTS { ?statement pq:P582 ?x }
  ?city p:P625/psv:P625 ?coords_value .
  ?coords_value wikibase:geoLatitude ?lat .
  ?coords_value wikibase:geoLongitude ?lon .
```

Ukázka 8: Rozšíření SPARQL dotazu o volitelné části

```
OPTIONAL {
  ?city rdfs:label ?citylabel .
  FILTER ( LANG(?citylabel) = "en" )
} OPTIONAL {
  ?cityarticle schema:about ?city;
  schema:isPartOf <https://en.wikipedia.org/> .
```

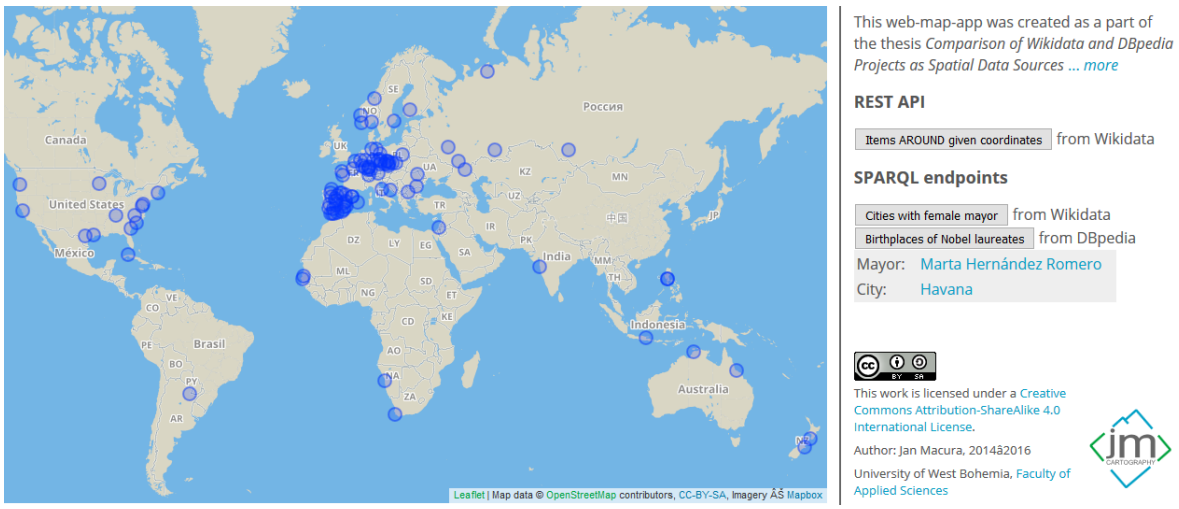
```
} OPTIONAL {
  ?mayor rdfs:label ?mayorlabel .
  FILTER ( LANG(?mayorlabel) = "en" )
} OPTIONAL {
  ?mayorarticle schema:about ?mayor;
  schema:isPartOf <https://en.wikipedia.org/> .
}}
```

Dotaz je poté potřeba zapsat jako řetězec v jazyce JavaScript a zakódovat jej do URI adresy, aby mohl být odeslán na SPARQL endpoint a vyhodnocen. V tomto případě jsou pro jednotlivé výsledky opět vytvořeny mapové značky a přidány do nové vrstvy v mapě, ale navíc je do jejich názvu uložena získaná informace o názvu města, odkazu na článek na Wikipedii o něm, jménu starostky a odkazu na článek o ní. Kód pro tento postup je shrnut v ukázce 9. Díky tomu lze pak tuto informaci znovu získat při kliknutí na některou ze značek v okně. Výsledek postupu je zobrazen na obrázku 10, v pravém panelu jsou vidět informace o aktivním prvku mapy – město Havana, starostka Marta Hernandéz Romero.

Ukázka 9: Získání a zpracování dat z WikidataQuery Service

```
var queryUrl = "https://query.wikidata.org/sparql?query=" +
  encodeURIComponent(query)+"&format=json";
$.ajax({
  dataType: "json",
  url: queryUrl,
  success: function(data) {
    var results = data.results.bindings;
    for (var i = 0; i < results.length; i++) {
      var mark = L.circleMarker([results[i].lat.value, results[i].
          lon.value], {radius: 7});
      var ml = results[i].mayorlabel ? results[i].mayorlabel.value
          : 'n/a';
      var ma = results[i].mayorarticle ? results[i].mayorarticle.
          value : null;
      var cl = results[i].citylabel ? results[i].citylabel.value :
          'n/a';
      var ca = results[i].cityarticle ? results[i].cityarticle.
          value : null;
      mark.name = 'wd|' + ml + '|' + ma + '|' + cl + '|' + ca;
      mark.on('click', showInfo);
      cities_layer.addLayer(mark);
    }
    cities_layer.addTo(map);
```

```
19 }});
```



Obrázek 10: Webová mapa zobrazující města, jež mají v čele ženu.

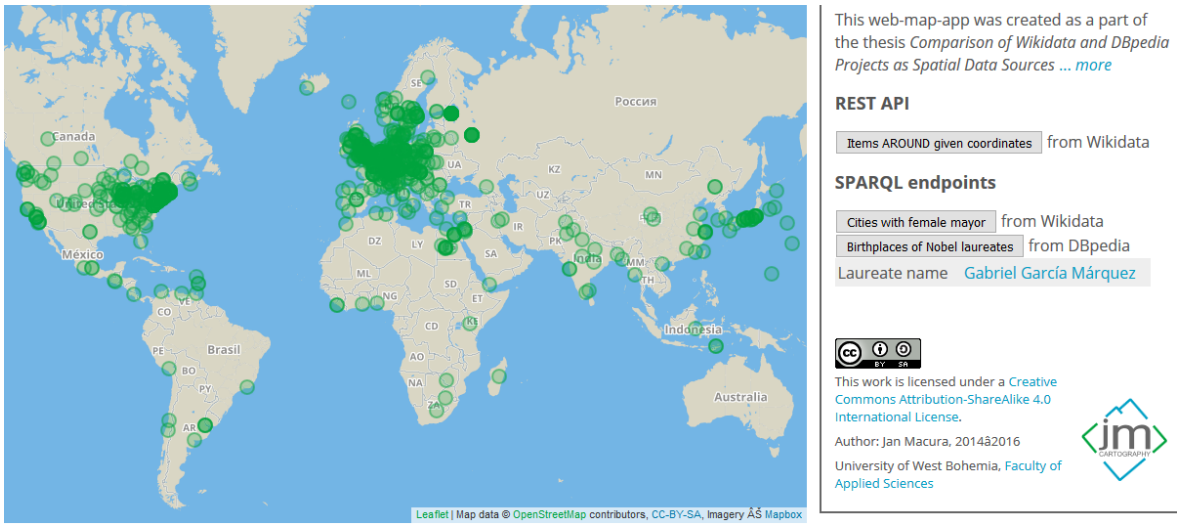
4.4.3 Dotaz přes DBpedia SPARQL endpoint

Rozhraním pro vzdálený přístup k datům v databázi DBpedia je opět SPARQL endpoint. Stejně jako v případě WikidataQuery Service, i zde je nejdůležitější správně formulovat SPARQL dotaz, k čemuž je potřeba dobře znát datový model a strukturu dat v databázi DBpedia. V tomto příkladu hledáme místa, ve kterých se narodili nositelé Nobelovy ceny (či Nobelových cen). Rozdíl oproti WikidataQuery Service je především v použitých vlastnostech a prefixech. Dále též zde využijeme agregačních funkcí a seskupení, pro eliminaci opakujících se souřadnic, jak již bylo rozebráno v kapitole 4.2.1. Kód celého SPARQL dotazu je na ukázce 10. Použití SPARQL dotazu a zpracování výsledků v JavaScriptu již probíhá stejně jako v případě databáze Wikidata. Jak je vidět na obrázku 11, po načtení bodů do mapy získáme tematickou mapu, která bude metodou teček zobrazovat prostorové rozložení rodných míst nositelů Nobelovy ceny.

Ukázka 10: SPARQL dotaz pro nalezení rodných míst nositelů Nobelovy ceny, z databáze DBpedia

```
1 PREFIX dct: <http://purl.org/dc/terms/>
  PREFIX dbc: <http://dbpedia.org/resource/Category:>
3 SELECT ?author SAMPLE(?lat) AS ?lat SAMPLE(?lon) AS ?lon SAMPLE(?wp
    ) AS ?authorarticle SAMPLE(?name) AS ?name WHERE {
   ?author dct:subject ?any .
5  ?any skos:broader+ dbc:Nobel_laureates .
   ?author a dbo:Person .
```

```
  ?author dbo:birthPlace ?place .
  ?place geo:lat ?lat .
  ?place geo:long ?lon .
  OPTIONAL {
    ?author prov:wasDerivedFrom ?wp .
  } OPTIONAL {
    ?author dbp:name ?name
    FILTER ( LANG(?name) = "en" ) .
}} GROUP BY (?author)
```



Obrázek 11: Webová mapa zobrazující rodná místa nositelů Nobelovy ceny.

Výsledkem celého postupu je dynamická webová stránka nazvaná LinkedMap[1] s mapovým oknem a postranním panelem, ze kterého lze spouštět jednotlivé akce. Webová aplikace jako celek může být šířena pod licencí CC-BY-SA 4.0. Zdrojové kódy souborů v HTML, CSS a JavaScriptu jsou přiloženy v příloze C a zároveň jsou k nalezení na serveru GitHub[2] pod licencí MPLv2, odkud je může kdokoli jednoduše zkopírovat a upravit pro vlastní aplikaci.

[1] http://home.zcu.cz/~jmacura/bp/
[2] https://github.com/jmacura/LinkedMap

5 Závěr

Cíli této práce bylo zjistit, jaká prostorová data existují v Linked Open Data projektech DBpedia a Wikidata a prozkoumat, jaká je jejich využitelnost pro aplikace. Nejprve bylo potřeba se seznámit s hlavními standardy Linked Data, jako jsou formáty RDF, OWL nebo dotazovací jazyk SPARQL. Poté bylo provedeno detailní prozkoumání datových modelů obou řešených databázových projektů.

V prvním kroku byla zjištěna a popsána struktura databáze DBpedia. Podrobně byl rozepsán vznik databáze DBpedia a způsob extrakce dat z encyklopedie Wikipedia, stejně jako rozdělení databáze podle jazyka zdrojové verze Wikipedie.

Ve druhém kroku byla zkoumána a následně zdokumentována struktura databáze Wikidata. Po rozsáhlé rešerši byl popsán datový model Wikibase DataModel a způsob jeho převedení do formy RDF. Taktéž bylo zjištěno, jak jsou v současnosti data z projektu Wikidata využívána v encyklopedii Wikipedia a jaké jsou vazby obou projektů, Wikidata i DBpedia, na ostatní Linked Open Data.

Poté byla provedena rešerše stávajících prací, které by se zabývaly srovnáváním znalostních bází či ontologií. Vzhledem k tomu, že nebyla nalezena žádná metodika, kterou by bylo možné snadno a efektivně aplikovat na problematiku prostorových dat, byl vytvořen vlastní postup porovnání obou databází z hlediska jejich celkové struktury, přístupnosti, způsobu popisu prostorových jevů a obsáhlosti.

Na základě sestaveného postupu bylo provedeno srovnání řešených projektů tak, aby jejich potencionální uživatel mohl jednoduše identifikovat rozdíly mezi nimi a porovnat jejich relativní vhodnost či nevhodnost pro zamýšlenou kartografickou aplikaci.

Dále byly analyzovány prostorové koncepty z obou datových sad. Podle dříve publikovaných prací byla ověřena existence jedné chyby v databázi DBpedia, kterou je duplicita souřadnic v oblasti Austrálie a byl nalezen způsob, jak se vypořádat s obdobným problémem, uváděním více souřadnic pro plošné jevy. Rovněž byla zdokumentována specifika ukládání prostorových dat v databázi Wikidata a s tím spojená rizika při prohledávání a vizualizaci dat.

V předposledním kroku byla provedena rešerše existujících aplikací, které pracují s prostorovými daty z některé z řešených databází.

Jako výstup práce byla vytvořena vlastní webová mapová aplikace, která vizualizuje data z projektů DBpedia i Wikidata, a byl podrobně popsán jeden z možných postupů, jak takovou aplikaci vytvořit. Zdrojový kód vytvořené aplikace je otevřený a dostupný na webu GitHub, aby jej kdokoliv mohl snadno prozkoumat a upravit pro svůj účel.

Na celou práci by bylo možné navázat zaměřením se na konkrétní problémy v některé z řešených znalostních bází, například jak zlepšit způsob extrakce dat z Wikipedie

tak, aby v databázi DBpedia nevznikaly duplicitní souřadnice, či jak v databázi Wikidata zavést podporu pro popis jevů i pomocí linií, polygonů či složitějších geometrických tvarů.

Seznam použité literatury

[1] Připěvatelé Wikipedie. Wikipedia. *Wikipedia, The Free Encyclopedia* [online]. 2014-12-22, 03:49 UTC [cit. 2014-12-22]. Dostupné z: http://en.wikipedia.org/w/index.php?title=Wikipedia&oldid=639134625

[2] Přispěvatelé Wikipedie. Nupedia. *Wikipedia, The Free Encyclopedia* [online]. 2014-11-01, 10:40 UTC [cit. 2014-12-22]. Dostupné z: http://en.wikipedia.org/w/index.php?title=Nupedia&oldid=631997088

[3] SCHREIBER, Guus, RAIMOND, Yves. *RDF 1.1 Primer: W3C Working Group Note 24 June 2014* [online]. MANOLA, Frank, MILLER, Eric, McBRIDE, Brian. World Wide Web Consortium, 2014-06-24 [cit. 2016-05-01]. Dostupné z: https://www.w3.org/TR/2014/NOTE-rdf11-primer-20140624/#section-data-model

[4] AUER, Sören, BIZER, Christian, KOBILAROV, Georgi, LEHMANN, Jens, CYGANIAK, Richard, IVES, Zachary. *DBpedia: A Nucleus for a Web of Open Data*. The Semantic Web. 2007. Volume 4825, 722735. ISSN 0302-9743. ISBN 978-3-540-76298-0. Dostupné z: http://www.cis.upenn.edu/~zives/research/dbpedia.pdf

[5] AUER, Sören, LEHMANN, Jens. What Have Innsbruck and Leipzig in Common? Extracting Semantics from Wiki Content. *The Semantic Web: Research and Applications*. Volume 4519. Innsbruck: 4th European Semantic Web Conference, 2007-06-03, s. 503–517. DOI 10.1007/978-3-540-72667-8_36. ISSN 0302-9743. ISBN 978-3-540-72667-8. Dostupné z: http://link.springer.com/chapter/10.1007/978-3-540-72667-8_36

[6] VRANDEČIĆ, Denny, KRÖTZSCH, Markus. Wikidata: A Free Collaborative Knowledgebase. *Communications of the ACM*. 2014-10, Vol. 57 No. 10, 7885. DOI 10.1145/2629489. Dostupné z: http://cacm.acm.org/magazines/2014/10/178785-wikidata/fulltext

[7] Přispěvatelé OpenStreetMap Wiki. Key:Wikidata. *OpenStreetMap Wiki* [online]. 2015-02-23, 17:43 UTC [cit. 2015-03-05]. Dostupné z: https://wiki.openstreetmap.org/wiki/Key:wikidata

[8] BERNERS-LEE, Tim. *Linked Data* [online]. 2006-07-27, poslední změna 2009-06-18 [cit. 2015-05-07]. Dostupné z: http://www.w3.org/DesignIssues/LinkedData.html

[9] HECHT, Brent, MOXLEY, Emily. Terabytes of Tobler: Evaluating the First Law

in a Massive, Domain-Neutral Representation of World Knowledge. *Spatial Information Theory*. Volume 5756. 9th International Conference, COSIT 2009 Aber Wrac'h, France, s. 88–105. DOI 10.1007/978-3-642-03832-7_6. ISBN 978-3-642-03832-7. ISSN 0302-9743. Dostupné z: http://link.springer.com/chapter/10.1007/978-3-642-03832-7_6

[10] ERXLEBEN, Fredo, GÜNTHER, Michael, KRÖTZSCH, Markus, MENDEZ, Julian, VRANDEČIĆ, Denny. Introducing Wikidata to the Linked Data Web. **In:** *Proceedings of the 13th International Semantic Web Conference*. Springer, 2014, s. 50–65. Dostupné z: http://korrekt.org/papers/Wikidata-RDF-export-2014.pdf

[11] BIZER, Christian, LEHMANN, Jens, KOBILAROV, Georgi, AUER, Sören, BECKER, Christian, CYGANIAK, Richard, HELLMANN, Sebastian. DBpedia - A crystallization point for the Web of Data. *Web Semantics: Science, Services and Agents on the World Wide Web*. 2009. Volume 7, Issue 3, s. 154–165. DOI 10.1016/j.websem.2009.07.002. ISSN 1570-8268. Dostupné z: http://www.sciencedirect.com/science/article/pii/S1570826809000225

[12] HOVY, Eduard. Comparing Sets of Semantic Relations in Ontologies. **In:** *Semantics of Relationships: An Interdisciplinary Perspective*. Kluwer, 2002, Chapter 6, s. 91–110. Dostupné z: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.68.972&rep=rep1&type=pdf

[13] VISSER, Pepijn R. S., BENCH-CAPON, Trevor J. M. A Comparison of Four Ontologies for the Design of Legal Knowledge Systems. *Artificial Intelligence and Law*. Kluwer Academic Publishers, 1998-03-01, Volume 6, Issue 1, s. 27–57. DOI 10.1023/A:1008251913710. ISSN 1572-8382. Dostupné z: http://link.springer.com/article/10.1023/A:1008251913710

[14] ČADA, Václav, MILDORF, Tomáš. Delimitation of reference geodata from land data model. **In:** *GIS Ostrava 2005*. Ostrava: VŠB - TUO, 2005. s. 1-12. Dostupné z: http://gis.vsb.cz/GIS_Ostrava/GIS_Ova_2005/Sbornik/en/Referaty/cada.pdf

[15] LUKOVNIKOV, Denis, STADLER, Claus, KONTOKOSTAS, Dimitris, HELLMANN, Sebastian, LEHMANN, Jens. DBpedia Viewer - An Integrative Interface for DBpedia Leveraging the DBpedia Service Eco System. **In:** *Linked Data on the Web 2014*. Seoul: CEUR Workshop Proceedings, 2014-04-08, Vol-1184. ISSN 1613-0073. Dostupné z: http://events.linkeddata.org/ldow2014/papers/ldow2014_paper_05.pdf

[16] MEUSEL, Robert, SPAHIU, Blerina, BIZER, Christian, PAULHEIM, He-

iko. Towards Automatic Topical Classification of LOD Datasets. **In:** *Linked Data on the Web 2015*. Florence: CEUR Workshop Proceedings, dosud nevydáno. ISSN 1613-0073. Dostupné z: http://events.linkeddata.org/ldow2015/papers/ldow2015_paper_03.pdf

[17] RAPANT, Petr. *Geoinformatika a geoinformační technologie*. 1. vydání. Ostrava: VŠB – Technická univerzita Ostrava, 2006. 513 s. ISBN 80-248-1264-9.

[18] ALASOUD, Ahmed, HAARSLEV, Volker, SHIRI, Nematollaah. An empirical comparison of ontology matching techniques. *Journal of Information Science*. 2009-03-24 (online), 2009-08 (print), vol. 35, no. 4, s. 379–397. DOI 10.1177/0165551508100383. Dostupné z: http://jis.sagepub.com/content/35/4/379

[19] SHI, Hui, MALY, Kurt, ZEIL, Steven, ZUBAIR, Mohammad. Comparison of Ontology Reasoning Systems Using Custom Rules. **In:** *Proceedings of the International Conference on Web Intelligence, Mining and Semantics*. New York: ACM, 2011, Article No. 16. DOI 10.1145/1988688.1988708. ISBN 978-1-4503-0148-0. Dostupné z: http://dl.acm.org/citation.cfm?id=1988708

[20] LEE, Chulki, PARK, Sungchan, LEE, Dongjoo, LEE, Jae-Won, JEONG, Ok-Ran, LEE, Sang-Goo. A comparison of ontology reasoning systems using query sequences. **In:** *Proceedings of the 2nd international conference on Ubiquitous information management and communication*. New York: ACM, 2008, s. 543–546. DOI 10.1145/1352793.1352907. ISBN 978-1-59593-993-7. Dostupné z: http://dl.acm.org/citation.cfm?id=1352907

[21] ISMAIL, Khairul Nurmazianna, BAKAR, Zainab Abu. Ontology Structure Comparison. **In:** *2013 IEEE Conference on e-Learning, e-Management and e-Services (IC3e)*. Kucing: IEEE, 2013, s. 148–151. DOI 10.1109/IC3e.2013.6735982. ISBN 978-1-4799-1573-6. http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6735982

[22] ISMAYILOV, Ali, KONTOKOSTAS, Dimitris, AUER, Sören, LEHMANN, Jens, HELLMANN, Sebastian. Wikidata through the Eyes of DBpedia. **In:** *ISWC 2015 Proceedings*. Bethlehem: CoRR, 2015, abs/1507.04180. Dostupné z: http://eis.iai.uni-bonn.de/upload/paper/ISWC2015dataonto_submission_4.pdf

[23] EGENHOFER, Max J., HERRING, John. A mathematical framework for the definition of topological relationships. **In:** *Fourth international symposium on spatial data handling*. Zurich: 1990, s. 803–813. Dostupné z: http://www.spatial.maine.edu/~max/MJEJRH-SDH1990.pdf

[24] BORST, Willem Nico. *Construction of engineering ontologies for knowledge sha-*

ring and reuse. 1997. Disertační práce. Universiteit Twente. Dostupné z: http://purl.utwente.nl/publications/17864

[25] STADLER, Claus, LEHMANN, Jens, HÖFFNER, Konrad, AUER, Sören. LinkedGeoData: A Core for a Web of Spatial Open Data. *Semantic Web Journal*. IOS Press, 2012, vol. 3, no. 4, s. 333–354. Dostupné z: http://svn.aksw.org/papers/2011/SWJ_LinkedGeoData/public.pdf

[26] CASTELLS SALA, Jordi. IFA: Defining Vague Places with Web Knowledge, a Semantic Approach. Enschede: 2012 [cit. 2015-11-26]. Universitatit Twente. Dostupné z: https://www.mendeley.com/research/defining-vague-places-web-knowledge-semantic-approach/

[27] PELLISSIER TANON, Thomas, VRANDEČIĆ, Denny, SCHAFFERT, Sebastian, STEINER, Thomas, PINTSCHER, Lidya. From Freebase to Wikidata: The Great Migration. **In:** *World Wide Web Conference 2016*. ACM, 2016. Dostupné z: http://static.googleusercontent.com/media/research.google.com/cs//pubs/archive/44818.pdf

[28] MOLLOY, Mark. The planet's history of violence over 4,000 years in one simple map. *The Telegraph* [online]. 2016-03-02 [cit. 2016-05-18] ISSN 0307-1235. Dostupné z: http://www.telegraph.co.uk/news/worldnews/12180516/Geography-of-violence-Map-records-every-battle-ever-fought.html

Seznam příloh

A Použité dotazy v jazyce SPARQL

A.1 Počet propojení mezi projekty Wikidata a OpenStreetMap

```
PREFIX wd: <http://www.wikidata.org/entity/>
SELECT Count(?osm) WHERE {
  { ?osm wd:P402s ?x . }
  UNION
  { ?osm wd:P1282s ?y .}
}
```

A.2 Počet odkazů z databáze Wikidata do databáze Freebase

```
PREFIX wd: <http://www.wikidata.org/entity/>
PREFIX p: <http://www.wikidata.org/prop/>
SELECT Count(?concept) WHERE {
  ?concept p:P646 ?x .
}
```

A.3 Počet položek se souřadnicemi v databázi Wikidata podle kosmického tělesa

```
PREFIX wikibase: <http://wikiba.se/ontology#>
SELECT (Count(?v) AS ?count) ?globe WHERE
{
  ?v wikibase:geoGlobe ?globe
}
GROUP BY ?globe
ORDER BY DESC(?count)
```

A.4 Počet prvků databáze DBpedia ve třídě yago:PermanentlyLocatedEntity

```
SELECT Count(?place) WHERE
{
  ?place a yago:YagoPermanentlyLocatedEntity .
}
```

A.5 Počet prvků databáze DBpedia ve třídě geo:SpatialThing

```
SELECT Count(?place) WHERE
{
  ?place a geo:SpatialThing .
}
```

A.6 Počet všech prvků se souřadnicemi v databázi DBpedia

```
SELECT Count(DISTINCT ?place) WHERE {{
  ?place dbp:latd ?lat .
  ?place dbp:longd ?lon .
} UNION {
  ?place dbp:latD ?lat .
  ?place dbp:longD ?lon .
} UNION {
  ?place geo:geometry ?wkt.
} UNION {
  ?place geo:lat ?lat .
  ?place geo:long ?lon .
} UNION {
  ?place georss:point ?point .
} UNION {
  ?place dbp:location ?loc .
} UNION {
  ?place dbo:location ?loc .
} UNION {
  ?place dbp:latns ?dir .
} UNION {
  ?place dbp:longew ?dir .
} UNION {
  ?place dbp:longDegrees ?lat .
  ?place dbp:latDegrees ?lon .
} UNION {
  ?place dbp:latDirection ?dir.
} UNION {
  ?place dbp:latNs ?dir .
} UNION {
  ?place dbp:coordinate ?coor .
} UNION {
  ?place dbp:latitude ?lat .
  ?place dbp:longitude ?lon .
}}
```

A.7 Prvky databáze DBpedia s vlastností geo:geometry

```
SELECT DISTINCT ?place SAMPLE(?wkt) AS ?wkt WHERE {
  ?place geo:geometry ?wkt.
}
GROUP BY ?place
```

A.8 Prvky databáze DBpedia s vlastnostmi geo:lat a geo:long

```
SELECT DISTINCT ?place SAMPLE(?lat) AS ?lat SAMPLE(?lon) AS ?lon
    WHERE {
  ?place geo:lat ?lat .
  ?place geo:long ?lon .
}
GROUP BY ?place
```

A.9 Prvky databáze DBpedia s vlastností grs:point

```
SELECT DISTINCT ?place SAMPLE(?point) AS ?point WHERE {
  ?place georss:point ?point.
}
GROUP BY ?place
```

A.10 Všechny prvky databáze Wikidata se souřadnicemi

```
PREFIX wd: <http://www.wikidata.org/entity/>
PREFIX wikibase: <http://wikiba.se/ontology#>
PREFIX p: <http://www.wikidata.org/prop/>
PREFIX psv: <http://www.wikidata.org/prop/statement/value/>
SELECT DISTINCT ?place ?lat ?lon WHERE {
  ?place p:P625/psv:P625 ?coords .
  ?coords wikibase:geoLatitude ?lat .
  ?coords wikibase:geoLongitude ?lon .
  ?coords wikibase:geoGlobe wd:Q2 .
}
```

A.11 Nejvyšší hory ve sluneční soustavě podle databáze Wikidata

```
PREFIX wikibase: <http://wikiba.se/ontology#>
PREFIX p: <http://www.wikidata.org/prop/>
PREFIX psv: <http://www.wikidata.org/prop/statement/value/>
PREFIX wdt: <http://www.wikidata.org/prop/direct/>
SELECT ?subj ?label ?coord ?elev (?qlabel AS ?units)
WHERE
{
  ?subj p:P2044 ?elev_s .
  ?elev_s psv:P2044 ?elev_v .
  ?elev_v wikibase:quantityUnit ?qu .
  ?elev_v wikibase:quantityAmount ?elev .
  FILTER(?elev > 8000) .
  ?subj wdt:P625 ?coord .
```

```
  ?subj rdfs:label ?label .
  FILTER(LANG(?label) = "en")
  OPTIONAL {
    ?qu rdfs:label ?qlabel .
    FILTER(LANG(?qlabel) = "en")
   }
} ORDER BY DESC(?elev)
```

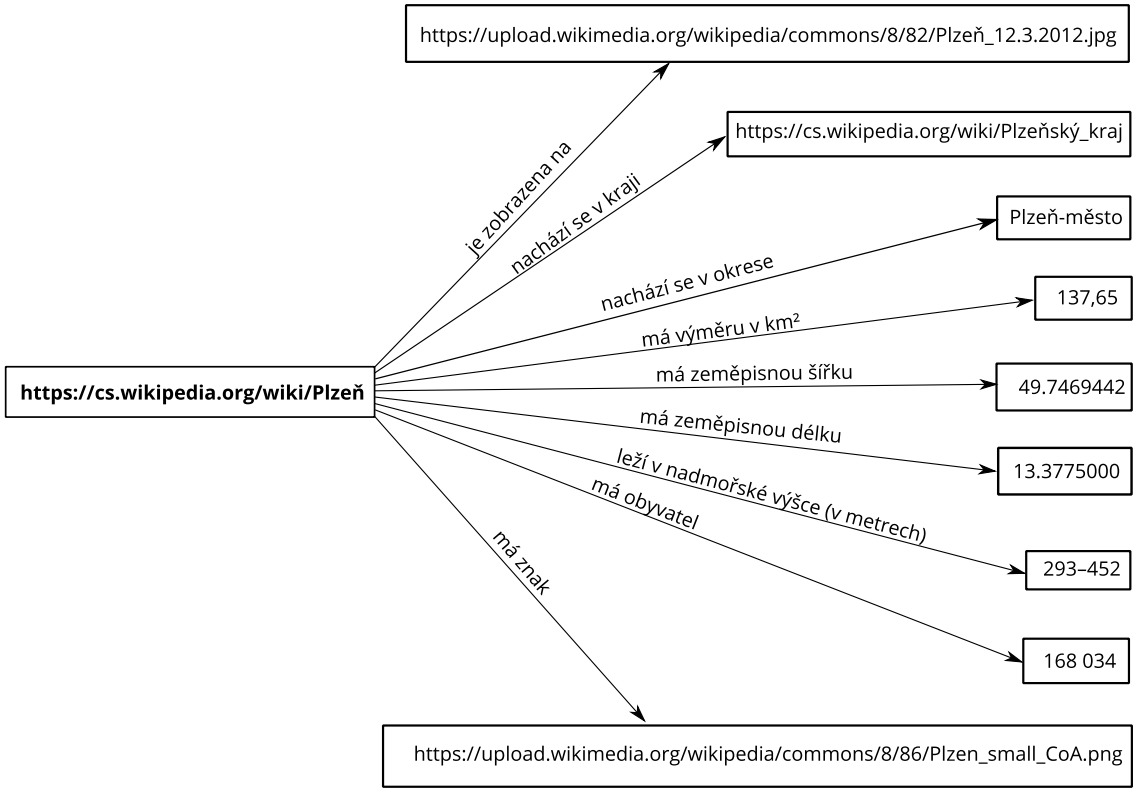
A.12 Města, která mají v čele ženu, podle databáze Wikidata

```
PREFIX wikibase: <http://wikiba.se/ontology#>
PREFIX : <http://www.wikidata.org/entity/>
PREFIX wdt: <http://www.wikidata.org/prop/direct/>
PREFIX p: <http://www.wikidata.org/prop/>
PREFIX ps: <http://www.wikidata.org/prop/statement/>
PREFIX psv: <http://www.wikidata.org/prop/statement/value/>
PREFIX pq: <http://www.wikidata.org/prop/qualifier/>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
SELECT DISTINCT ?lat ?lon ?citylabel ?cityarticle ?mayorlabel ?
    mayorarticle WHERE {
  ?city wdt:P31/wdt:P279* :Q515 .
  ?city p:P6 ?statement .
  ?statement ps:P6 ?mayor .
  ?mayor wdt:P21 :Q6581072 .
  FILTER NOT EXISTS { ?statement pq:P582 ?x }
  ?city p:P625/psv:P625 ?coords_value .
  ?coords_value wikibase:geoLatitude ?lat .
  ?coords_value wikibase:geoLongitude ?lon .
  OPTIONAL {
    ?city rdfs:label ?citylabel .
    FILTER ( LANG(?citylabel) = "en" )
  } OPTIONAL {
    ?cityarticle schema:about ?city;
    schema:isPartOf <https://en.wikipedia.org/> .
  } OPTIONAL {
    ?mayor rdfs:label ?mayorlabel .
    FILTER ( LANG(?mayorlabel) = "en" )
  } OPTIONAL {
    ?mayorarticle schema:about ?mayor;
    schema:isPartOf <https://en.wikipedia.org/> .
}}
```

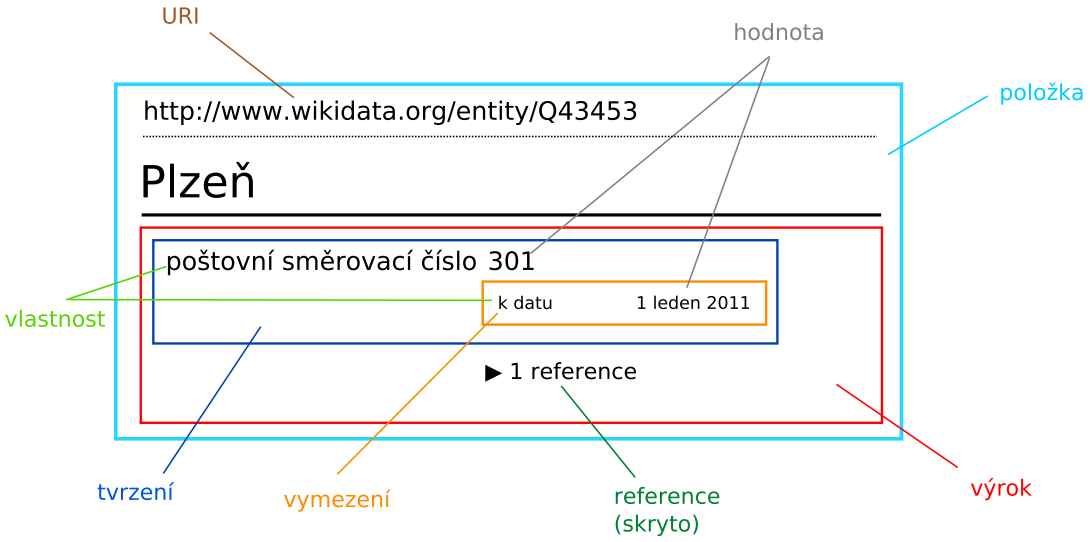
A.13 Rodná místa nositelů Nobelovy ceny, podle databáze DBpedia

```
PREFIX dct: <http://purl.org/dc/terms/>
PREFIX dbc: <http://dbpedia.org/resource/Category:>
SELECT ?author SAMPLE(?lat) AS ?lat SAMPLE(?lon) AS ?lon SAMPLE(?wp
    ) AS ?authorarticle SAMPLE(?name) AS ?name WHERE {
  ?author dct:subject ?any .
  ?any skos:broader+ dbc:Nobel_laureates .
  ?author a dbo:Person .
  ?author dbo:birthPlace ?place .
  ?place geo:lat ?lat .
  ?place geo:long ?lon .
  OPTIONAL {
    ?author prov:wasDerivedFrom ?wp .
  } OPTIONAL {
    ?author dbp:name ?name
    FILTER ( LANG(?name) = "en" ) .
}} GROUP BY (?author)
```

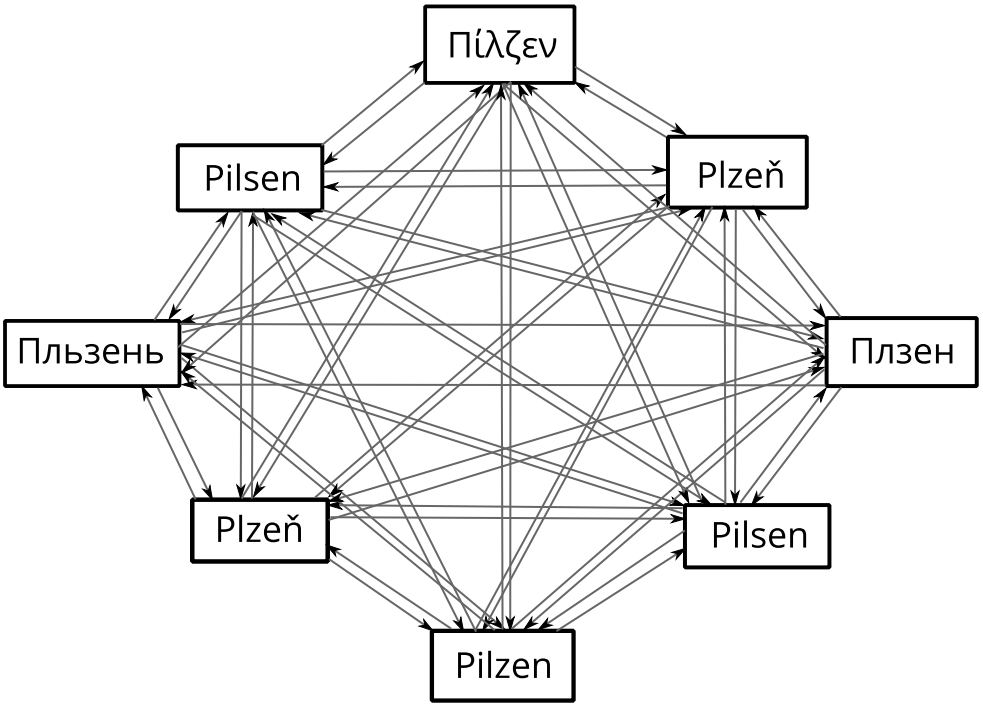
B Ilustrace ve vysokém rozlišení



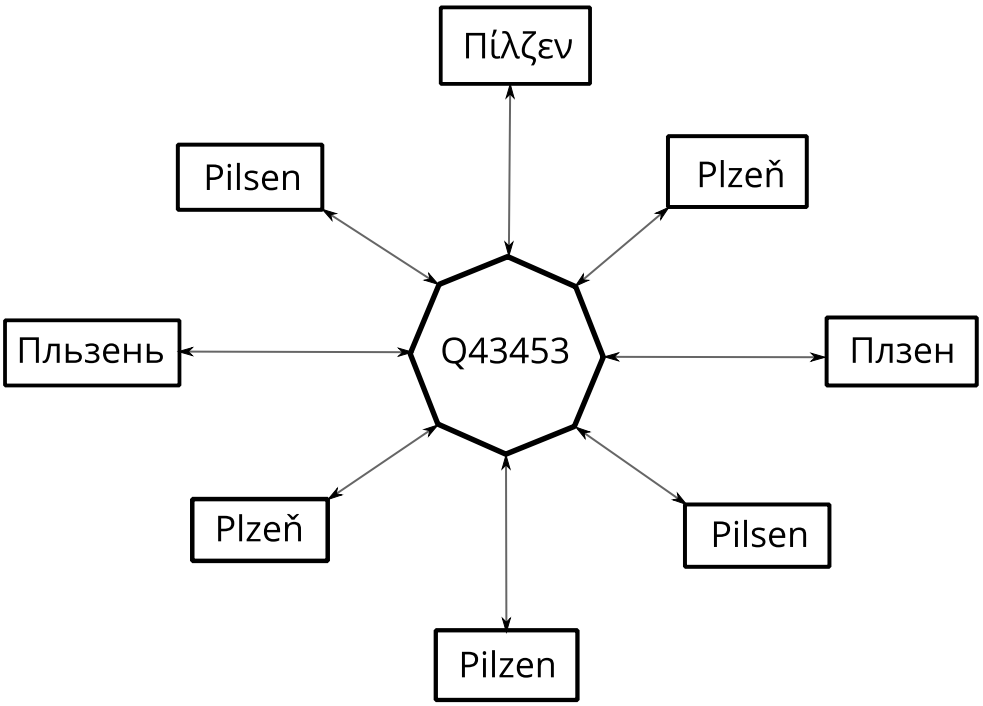
Obrázek 2: Reprezentace dat extrahovaných z Infoboxu na Wikipedii do formátu RDF.



Obrázek 3: Struktura položky databáze Wikidata.

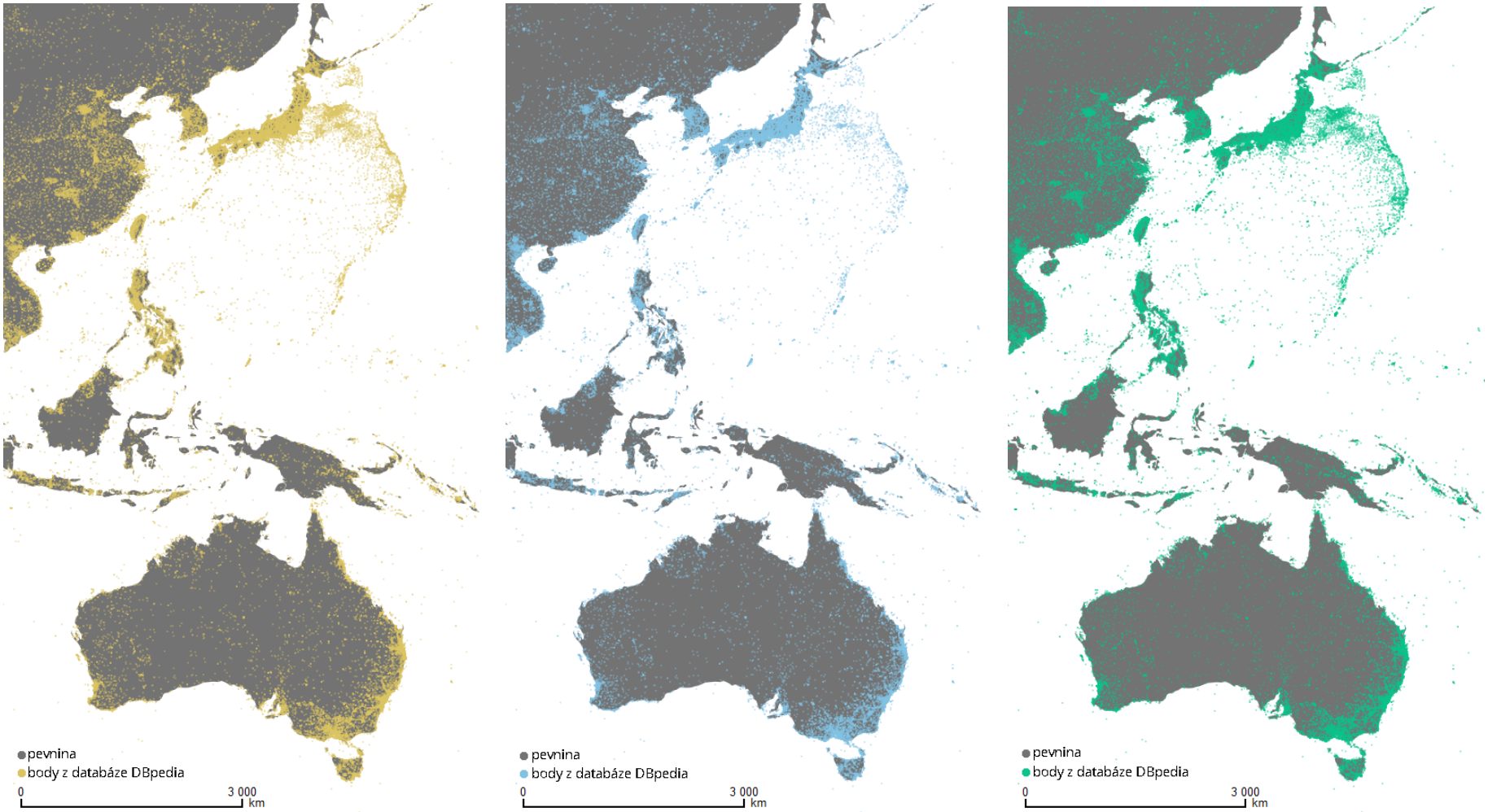


(a) Schéma vzájemného propojení článků Wikipedie před zavedením databáze Wikidata.



(b) Schéma vzájemného propojení článků Wikipedie po zavedení databáze Wikidata.

Obrázek 5: Schéma vzájemného propojení článků Wikipedie před a po zavedení databáze Wikidata.

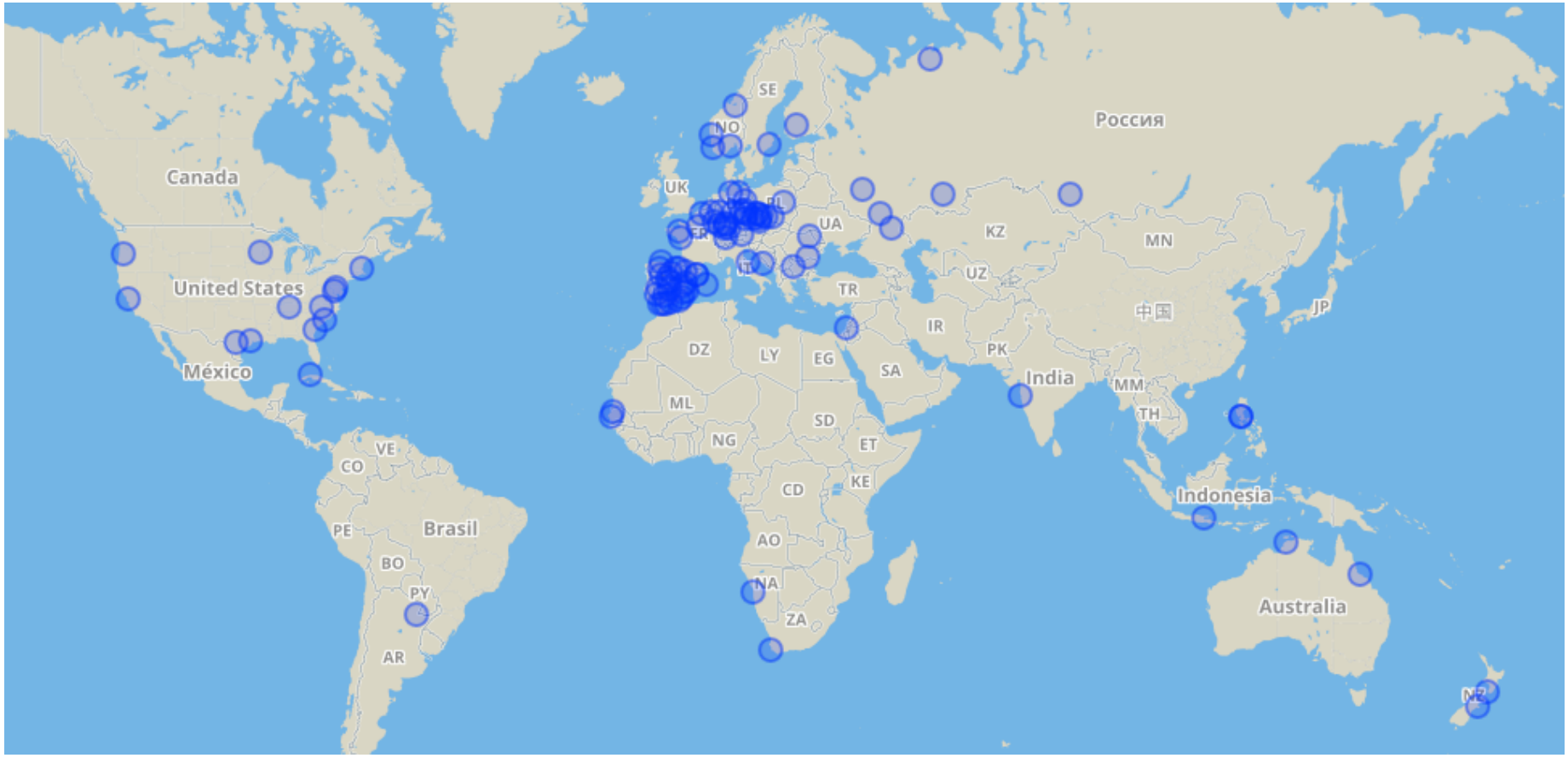


(a) Vlastnosti `geo:lat` a `geo:long`.

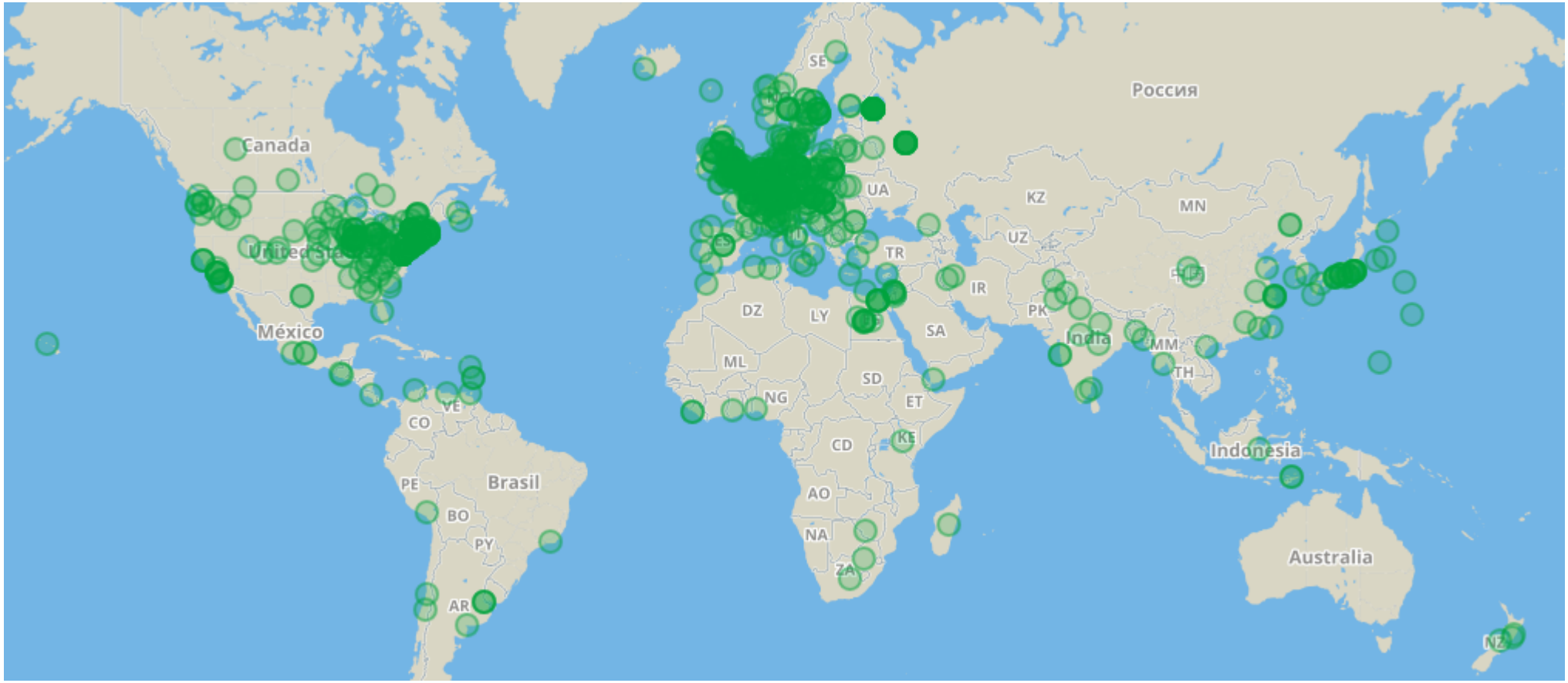
(b) Vlastnost `grs:point`.

(c) Vlastnost `geo:geometry`.

Obrázek 7: Zobrazení zrcadlové Austrálie u bodů se souřadnicemi z databáze DBpedia, podle ovlivněných vlastností.



Obrázek 10: Vizualizovaný výsledek dotazu „Najdi města, jež mají v čele ženu“.



Obrázek 11: Vizualizovaný výsledek dotazu „Najdi rodná místa nositelů Nobelovy ceny“.

C Zdrojové kódy programů

C.1 LinkedMap index.html

```
<!doctype html>
<html dir="ltr" lang="cs">
<head>
  <meta charset="UTF-8">
  <meta name="author" content="jmacura 2014-2016">
  <meta name="robots" content="index, follow">
  <meta http-equiv="content-type" content="text/html; charset=utf-8
     ">
  <title>LinkedMap -- DBpedia and Wikidata visualizations</title>
  <link rel="stylesheet" href="http://cdn.leafletjs.com/leaflet
     -0.7.3/leaflet.css" type="text/css" media="all">
  <link rel="stylesheet" href="styles.css" type="text/css" media="
     all">
  <script src="http://cdn.leafletjs.com/leaflet-0.7.3/leaflet.js"><
     /script>
  <script src="http://ajax.googleapis.com/ajax/libs/jquery/1.11.2/
     jquery.min.js"></script>
  <script src="linkedmap.js" charset="utf-8"></script>
</head>

<body lang="cs">
<main>
<div id="map"></div>

<aside>
<div id="content">
  <h1>LinkedMap</h1>
  <p>This web-map-app was created as a part of the thesis <em>
     Comparison of Wikidata and DBpedia Projects as Spatial Data
     Sources</em> <a href="#" onclick="aboutHider();" class="hidden
     " style="display: inline;"> ... <em>more</em></a><span class="
     hidden">at <a href="http://fav.zcu.cz">Faculty of Applied
     Sciences</a>, <a href="http://zcu.cz">UWB</a>.
  The point of this geoapp is to present some ways, how to mine
     spatial data from <a href="http://wikidata.org">Wikidata</a>
     and <a href="http://dbpedia.org">DBpedia</a> projects.
  For more information about the projects themself and spatial data
      they contain, please refer the thesis itself. <a href="#"
     onclick="aboutHider();"><em>less</em></a></span></p>
  <h2>REST API</h2>
```

```
27    <button onclick="aroundQuery();">Items AROUND given coordinates</
          button> from Wikidata<br>
28    <h2>SPARQL endpoints</h2>
29    <button onclick="wdQuery();">Cities with female mayor</button>
          from Wikidata<br>
30    <button onclick="dbpQuery();">Birthplaces of Nobel laureates</
          button> from DBpedia
31    <div id="info-block"><p> <p></div>
32    <footer>
33      <div class="logo">
34        <img src="logo.svg" alt="author's logo">
35      </div>
36      <div class="credits">
37        <a rel="license" href="http://creativecommons.org/licenses/by
              -sa/4.0/"><img alt="Creative Commons License" style="
              border-width:0" src="https://i.creativecommons.org/l/by-sa
              /4.0/88x31.png"></a><br>
38        This work is licensed under a <a rel="license" href="http://
              creativecommons.org/licenses/by-sa/4.0/">Creative Commons
              Attribution-ShareAlike 4.0 International License</a>
39        and the source code is licensed under a <a rel="license" href
              ="https://www.mozilla.org/en-US/MPL/2.0/">Mozilla Public
              License Version 2.0</a>.
40        <p>Author: Jan Macura, 2014--2016</p>
41        <p>University of West Bohemia, <a href="http://www.fav.zcu.cz
              /en">Faculty of Applied Sciences</a></p>
42      </div>
43    </footer>
44  </div>
45  </aside>
46
47  <script>
48  if (L.Browser.ie6) {
49    alert('Internet Explorer 6 is not supported.');
50  }
51
52  var map = L.map('map').setView([49.77, 13.38], 12);
53  L.tileLayer('http://{s}.tiles.mapbox.com/v3/yjm.j5j87g72/{z}/{x}/{y
        }.png', {
54    attribution: 'Map data &copy; <a href="http://openstreetmap.org">
          OpenStreetMap</a> contributors, <a href="http://
          creativecommons.org/licenses/by-sa/2.0/">CC-BY-SA</a>, Imagery
           (c) <a href="http://mapbox.com">Mapbox</a>',
55    maxZoom: 18
```

```
56 }).addTo(map);
57 controller.addTo(map);
58
59 </script>
60
61 </main>
62 </body>
63 </html>
```

C.2 LinkedMap linkedmap.js

```
1  // global variables
2  var controller = L.control.layers(null, null); //control pane for
       layers, empty when started
3  var layerList = [false, false, false]; //this list ensures that
       every layer will exists only once
4
5  // This serves for some candy-eye effects in the main page
6  function aboutHider() {
7          var objs = document.getElementsByClassName("hidden");
8          for(var i = 0; i < objs.length; i++) {
9                  objs[i].style.display = (objs[i].style.display == "
                       inline") ? "none" : "inline";
10         }
11 }
12
13 // Currently unused
14 function onMapClick(e) {
15         alert("You clicked the map at " + e.latlng);
16 }
17
18 // **** This is the AROUND query ****
19 function getList(target) {
20         if(layerList[0]) {alert('Layer already exists'); return;}
21         $.getJSON(target + '&callback=?', function(data) { //jQuery
               function
22                 var d = data.props[625]; //only property P625 (
                       geographic coordinates) is interesting here
23                 //console.log(d);
24                 var wd_layer = L.featureGroup(null);
25                 for(var i = 0; i < d.length; i++) { //for each item
                        found create a pin-like marker and add it to
                       separate layer
26                         var coords = d[i][2].split("|"); //the
                               coordinates are written in form latitude
                               |longitude
27                         var mark = L.marker([coords[0],coords[1]]);
28                         wd_layer.addLayer(mark);
29                 }
30                 wd_layer.addTo(map); //add the layer of all newly
                       created markers to the map window
31                 controller.addOverlay(wd_layer, "Items around given
                        coordinates");
```

```
32                  map.setView([49.77, 13.38], 12); //zoom the map
                        window to the center of points
33                  layerList[0] = true; //prevent creating the layer
                        more than once
34          });
35  }
36
37  // **** This is binded with AROUND query ****
38  /* If the user confirms the content of the query, the getList()
        function is called */
39  function aroundQuery() {
40          if (confirm("Bude odeslán následující dotaz:\r\n\r\nAROUND
                [49.7462414,13.3778353,2]")) {
41                  getList("https://wdq.wmflabs.org/api?q=around%5B625
                        ,49.7462414,13.3778353,2%5D&props=625");
42          }
43  }
44
45  // Error logging
46  function logErr() {
47          console.log("Operation was not successful from unknown
                reason.");
48  }
49
50  // **** Background support for displaying info ****
51  function showInfo(e) { //e is the event, which called this function
52          var feature = e.target;
53          var infoBlock = document.getElementById("info-block");
54          //parsing values for info-block
55          var attrs = feature.name.split('|');
56          var ls, r, d, l, t;
57          //1st line
58          ls = document.createElement("TABLE");
59          r = document.createElement("TR");
60          d = document.createElement("TD");
61          t = document.createTextNode( (attrs[0] == "wd") ? "Mayor:"
                : "Laureate name");
62          d.appendChild(t); r.appendChild(d);
63          d = document.createElement("TD");
64          t = document.createTextNode(attrs[1]);
65          if (attrs[2] != "null") { //if the link to Wikipedia
                article about the person was retieved then create a link
                 element
66                  l = document.createElement("A");
```

```
67                  l.setAttribute("href", (attrs[0] == "dbp") ? attrs
                        [2].split('?')[0] : attrs[2]); //DBpedia
                        provides permanent links to older versions of
                        article, which are extended with the '?oldid=nnn
                        '
68                  l.setAttribute("target", "_blank");
69                  l.appendChild(t); d.appendChild(l);
70          }
71          else {d.appendChild(t);}
72          r.appendChild(d); ls.appendChild(r);
73          //2nd line
74          if (attrs[0] == "wd") { //only add another line, if the
                marker is about mayor
75                  r = document.createElement("TR");
76                  d = document.createElement("TD");
77                  t = document.createTextNode("City:");
78                  d.appendChild(t); r.appendChild(d);
79                  d = document.createElement("TD");
80                  t = document.createTextNode(attrs[3]);
81                  if (attrs[4] != "null") {//if the link to Wikipedia
                        article of the city was retrieved then create a
                        link element
82                      l = document.createElement("A"); l.
                            setAttribute("href", attrs[4]); l.
                            setAttribute("target", "_blank");
83                      l.appendChild(t); d.appendChild(l);
84                  }
85                  else {d.appendChild(t);}
86                  r.appendChild(d); ls.appendChild(r);
87          }
88          while(infoBlock.hasChildNodes()) { //remove existing
                information
89                  infoBlock.removeChild(infoBlock.firstChild);
90          }
91          infoBlock.appendChild(ls);
92  }
93
94  // **** this is for query female mayors from WikidataQuery Service
        ****
95  function wdQuery() {
96          if(layerList[1]) {alert('Layer already exists'); return;}
97          var cities_layer = L.featureGroup(null);
98          var url = "https://query.wikidata.org/sparql";
99          var query =
```

```
100                 'PREFIX wikibase: <http://wikiba.se/ontology#>\n' +
101                 'PREFIX : <http://www.wikidata.org/entity/>\n' +
102                 'PREFIX wdt: <http://www.wikidata.org/prop/direct
                        />\n' +
103                 'PREFIX p: <http://www.wikidata.org/prop/>\n' +
104                 'PREFIX ps: <http://www.wikidata.org/prop/statement
                        />\n' +
105                 'PREFIX psv: <http://www.wikidata.org/prop/
                        statement/value/>\n' +
106                 'PREFIX pq: <http://www.wikidata.org/prop/qualifier
                        />\n' +
107                 'PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema
                        #>\n' +
108                 'SELECT DISTINCT ?lat ?lon ?citylabel ?cityarticle
                        ?mayorlabel ?mayorarticle WHERE {\n' +
109                 '?city wdt:P31/wdt:P279* :Q515 . # instanceOf city
                        or something that is a subclass of city\n' +
110                 '?city p:P6 ?statement . # headOfGovernment
                        statement \n' +
111                 '?statement ps:P6 ?mayor . # the value of statement
                        \n' +
112                 '?mayor wdt:P21 :Q6581072 . # the sexOrGender of
                        the mayor is female\n' +
113                 'FILTER NOT EXISTS { ?statement pq:P582 ?x } #
                        there is no endDate qualifier\n' +
114                 '?city p:P625/psv:P625 ?coords_value . #
                        coordinates value for city\n' +
115                 '?coords_value wikibase:geoLatitude ?lat .\n' +
116                 '?coords_value wikibase:geoLongitude ?lon .\n' +
117                 'OPTIONAL { # optionally find English label(s) for
                        each city\n' +
118                       '?city rdfs:label ?citylabel .\n' +
119                       'FILTER ( LANG(?citylabel) = \"en\" )\n' +
120                 '} OPTIONAL {# optionally find Wikipedia article
                        for each city\n' +
121                       '?cityarticle schema:about ?city;\n' +
122                       'schema:isPartOf <https://en.wikipedia.org
                              /> .\n' +
123                 '} OPTIONAL {# optionally find English label(s) for
                         each mayor\n' +
124                       '?mayor rdfs:label ?mayorlabel .\n' +
125                       'FILTER ( LANG(?mayorlabel) = \"en\" )\n' +
126                 '} OPTIONAL {# optionally find Wikipedia article
                        for each mayor\n' +
```

```
127                         '?mayorarticle schema:about ?mayor;\n' +
128                         'schema:isPartOf <https://en.wikipedia.org
                                /> .\n' +
129                 '}}';
130         console.log(query); //show the SPARQL query in the console
131         var queryUrl = url+"?query="+encodeURIComponent(query)+"&
                format=json";
132         $.ajax({ //jQuery function, slightly more complex than
                getJSON()
133                 dataType: "json",
134                 url: queryUrl,
135                 success: function(data, stat) { //this function
                        will be called after successful end of AJAX
                        request
136                         //console.log(data);
137                         var results = data.results.bindings;
138                         for (var i = 0; i < results.length; i++) {
                                //create a pretty marker for each
                                retrieved pair of coordinates
139                                 var mark = L.circleMarker([results[
                                        i].lat.value, results[i].lon.
                                        value], {radius: 7});
140                                 var ml = results[i].mayorlabel ?
                                        results[i].mayorlabel.value : 'n
                                        /a';
141                                 var ma = results[i].mayorarticle ?
                                        results[i].mayorarticle.value :
                                        null;
142                                 var cl = results[i].citylabel ?
                                        results[i].citylabel.value : 'n/
                                        a';
143                                 var ca = results[i].cityarticle ?
                                        results[i].cityarticle.value :
                                        null;
144                                 mark.name = 'wd|' + ml + '|' + ma +
                                         '|' + cl + '|' + ca; //all
                                        retieved info is stored in the
                                        name of the marker for future
                                        use
145                                 mark.on('click', showInfo); // this
                                         binds an onClick event handler
                                        for each marker, so the info can
                                         be shown to the user
146                                 cities_layer.addLayer(mark);
```

```
147                         }
148                         cities_layer.addTo(map);
149                         controller.addOverlay(cities_layer, "Cities
                                with female mayor");
150                         map.setView([30, 0], 2); //zoom out the map
                                window to the whole world
151                         layerList[1] = true; //prevent creating
                                this layer more than once
152                 },
153                 error: function(jqXHR, status, error) { //if
                        something fails, log what is wrong
154                         console.log("status: ", status, " error: ",
                                error);
155                 }
156         }).fail(logErr);
157 }
158
159 // **** this is for query Nobel prize laureates from DBpedia SPARQL
        endpoint****
160 function dbpQuery() {
161         if(layerList[2]) {alert('Layer already exists'); return;}
162         var birthplace_layer = L.featureGroup(null);
163         var url = "http://dbpedia.org/sparql";
164         var query =
165                 'SELECT ?author SAMPLE(?lat) AS ?lat SAMPLE(?lon)
                        AS ?lon SAMPLE(?wp) AS ?authorarticle SAMPLE(?
                        name) AS ?authorname WHERE {\n' +
166                 '?author a dbo:Person . # firstly select persons (
                        we do not want institutions here)\n' +
167                 '?author dct:subject ?any . # then look what
                        classes they belong into ...\n' +
168                 '?any skos:broader+ dbc:Nobel_laureates . # ... and
                         choose NobelLaureates or their subclasses\n' +
169                 '?author dbo:birthPlace ?place . # get the
                        birtplace for each person\n' +
170                 '?place geo:lat ?lat . # latitude of birthplace\n'
                        +
171                 '?place geo:long ?lon . # longitude of birthplace\n
                        ' +
172                 'OPTIONAL { # optionally find the Wikipedia article
                         connected with each person\n' +
173                         '?author prov:wasDerivedFrom ?wp .\n' +
174                 '} OPTIONAL { # optionally find the English label(s
                        ) for each person\n' +
```

```
175                         '?author dbp:name ?name\n' +
176                         'FILTER ( LANG(?name) = "en" ) .\n' +
177                 '}} GROUP BY (?author)';
178         console.log(query);
179         var queryUrl = url+"?query="+encodeURIComponent(query)+"&
                format=json&callback=?";
180         $.ajax({ //jQuery function, slightly more complex than
                getJSON()
181                 dataType: "json",
182                 url: queryUrl,
183                 success: function(data, stat) { //this function
                        will be called after successful end of AJAX
                        request
184                         //console.log(data);
185                         var results = data.results.bindings;
186                         for (var i = 0; i < results.length; i++) {
                            //create a pretty marker for each
                            retrieved pair of coordinates
187                             var mark = L.circleMarker([results[
                                i].lat.value, results[i].lon.
                                value], {radius: 7, color: 'rgb
                                (0, 163, 61)'});
188                             var name = results[i].authorname ?
                                results[i].authorname.value : 'n
                                /a';
189                             var link = results[i].authorarticle
                                 ? results[i].authorarticle.
                                value : null;
190                             mark.name = 'dbp|' + name + '|' +
                                link; //all retieved info is
                                stored in the name of the marker
                                 for future use
191                             mark.on('click', showInfo); // this
                                 binds an onClick event handler
                                for each marker, so the info can
                                 be shown to the user
192                             birthplace_layer.addLayer(mark);
193                         }
194                         birthplace_layer.addTo(map);
195                         controller.addOverlay(birthplace_layer, "
                            Birthplaces of Nobel laureates");
196                         map.setView([30, 0], 2); //zoom out the map
                             window to the whole world
197                         layerList[2] = true; //prevent creating
```

```
                                this layer more than once
198                 },
199                 error: function(jqXHR, status, error) { // if
                        something fails, log what is wrong
200                         console.log("status: ", status, " error: ",
                                error);
201                 }
202         }).fail(logErr);
203 }
204
205
206 function showAnotherInfo(e) {
207         console.log(e.target);
208 }
```

D Obsah přiloženého CD

- latex/ – adresář se zdrojovým textem práce ve formátu LaTeX

  media/ – adresář s ilustracemi textu

    svg/ – adresář s vektorovými verzemi obrázků

  `macura_bp.tex` – zdrojový dokument LaTeX

- LinkedMap/ – adresář se zdrojovými soubory mapové aplikace *LinkedMap*
- LODIC/ – adresář se zdrojovými soubory programu *LOD Ion Cannon*
- spatialData – adresář s CSV soubory prostorových dat stažených z projektů Wikidata a DBpedia
- `macura_bp.pdf` – text bakalářské práce
- `zadani_jm.pdf` – oficiální zadání práce