Vysoká škola ekonomická v Praze

Fakulta informatiky a statistiky
Katedra informačního a znalostního inženýrství



Studijní program: Aplikovaná informatika
Obor: Informatika

Návrh datových struktur s využitím jazyka RDFS/OWL

BAKALÁŘSKÁ PRÁCE

Student : Linda Horáková

Vedoucí : Ing. Marek Dudáš

Oponent : Mgr. Jindřich Mynarz

2015

Prohlášení

Prohlašuji, že jsem bakalářskou práci zpracovala samostatně a že jsem uvedla všechny použité prameny a literaturu, ze které jsem čerpala.

V Praze dne 5. května 2015 . . . . . .

Linda Horáková

Poděkování

Ráda bych poděkovala Ing. Markovi Dudášovi za odborné vedení, ochotu, cenné rady a připomínky při zpracování této práce.

Abstrakt

Cílem bakalářské práce je prozkoumat možnosti a výhody publikování statistických dat na Sémantickém webu v souladu s principy Linked Data. V úvodní teoretické části jsou čtenáři seznámeni se základními vlastnostmi Sémantického webu, principy Linked Data a možnostmi publikování propojených dat. Hlavní náplní bakalářské práce je analýza existujících slovníků pro popis statistických dat, zejména pak v oblasti veřejných rozpočtů. Výstupem této analýzy je tvrzení, že pro popis statistických dat je vhodné používat obecné slovníky, jejichž možnosti jsou širší než u proprietárních ontologií. Výsledky analýzy byly ověřeny při experimentálním návrhu datové struktury pro oblast veřejných výdajů, která je postavena na obecném slovníku Data Cube Vocabulary. Součástí praktické části práce je také konstrukce SPARQL dotazu pro automatický převod existujících dat do podoby, která vyhovuje navržené datové struktuře.

Klíčová slova

Linked Data, Sémantický Web, datové struktury, Data Cube Vocabulary, ontologie

Abstract

The main aim of the bachelor thesis is to explore possibilities and advantages of publishing statistical data on Semantic Web according to Linked Data Principles. In theoretical part readers are familiarized with basic features of Semantic Web, principles and options of publishing Linked Data. The thesis is mainly focused on an analysis of existing vocabularies used for description of statistical data, especially in the field of public spending. The outcome of the analysis is the statement that general vocabularies are more suitable for description of statistical data, because they provide wider possibilities than proprietary ontologies. The result of the analysis was verified by the draft concept of data structure describing the field of public spending, which is based on general Data Cube Vocabulary. The practical part of the thesis also contains construction of SPARQL query which can be used for automatic transformation of current data to the shape that is suitable to the new data structure.

Keywords

Linked Data, Semantic Web, data structures, Data Cube Vocabulary, ontology

Obsah

1 Úvod 1

2 Sémantický web a principy Linked Data 2
2.1 Sémantický web 2
2.1.1 Vlastnosti Sémantického webu 3
2.1.2 Architektura Sémantického webu 4
2.2 Linked Data 6
2.2.1 Principy Linked Data 6
2.2.2 Výhody propojených dat 8
2.2.3 Linked Open Data 9
2.3 Reprezentace dat pomocí RDF a souvisejících technologií 11
2.3.1 Resource Description Framework 12
2.3.2 RDF Schema 14
2.3.3 Web Ontology Language 17
2.4 Data na Sémantickém webu 18
2.4.1 Editace dat pro Sémantický web 18
2.4.2 Publikování propojených dat 20
2.4.3 Práce s daty na Sémantickém webu 26

3 Analýza slovníků Sémantického webu pro statistická data 27
3.1 Data Cube Vocabulary 27
3.1.1 Architektura Data Cube Vocabulary 28
3.1.2 Doporučený postup návrhu datové struktury pomocí Data Cube Vocabulary 31
3.1.3 Implementace Data Cube Vocabulary 33
3.1.4 Programy pro práci se statistickými daty 34
3.2 Proprietární slovníky pro popis statistických dat 35

4 Návrh datové struktury pro data z oblasti veřejných rozpočtů 37
4.1 Ontologie Public Spending 37
4.2 Návrh datové struktury – Public Spending Data Cube 40
4.2.1 Volba jmenného prostoru a prefixu 41
4.2.2 Datová struktura 41
4.2.3 Ukázková data 45
4.3 Generování dat z existujících zdrojů pomocí SPARQL 49
4.3.1 Práce v prostředí Sesame a návrh SPARQL dotazu 50
4.4 Zpracování statistických dat v nástroji CubeViz 53

Závěr 55

Terminologický slovník 56

Seznam literatury 58

Seznam obrázků a tabulek 61

Základní text 61

Seznam obrázků 61

Seznam tabulek 61

Rejstřík 62

Příloha A: Public Spending Data Cube 63

Příloha B: Ukázková data pro Public Spending Data Cube 67

1 Úvod

S rostoucím množstvím dat publikovaných na Webu roste také potřeba s těmito údaji efektivně pracovat. Na klasickém „Webu dokumentů“ se informace vyskytují převážně v nestrukturované formě, není tedy povětšinou možné takové informace zpracovávat automaticky (pomocí strojů). Právě strojová čitelnost dat je důležitým aspektem v jejich využitelnosti. Za tímto účelem vzniklo rozšíření klasického World Wide Webu nazvané Sémantický web. Data na sémantickém webu jsou ukládána ve strojově čitelné formě pomocí formátu RDF, jsou tedy automaticky nalezitelná a zpracovatelná aplikacemi. Důležitou součástí Sémantického webu je oblast nazvaná Linked Data. Za tímto názvem se skrývá množina technologií a postupů, podle nichž lze publikovat data na Sémantickém webu tak, že jsou vzájemně propojována odkazy, což umožňuje efektivně vyhledávat související zdroje. To vede k tomu, že data mohou být využita i pro jiné účely, pro které byla původně publikována. Díky Linked Data jsou data ukládána do tzv. „*globálního prostoru dat*“. Tato práce se zaměřuje na publikování statistických dat. Pro popis dat tohoto typu existuje slovník Data Cube Vocabulary, který se doporučuje používat pro svou schopnost vyjádření multidimenzionality a vytváření různých pohledů do znalostní báze. Existuje také několik analytických nástrojů, které dovedou data popsaná tímto slovníkem zpracovávat.

Cílem bakalářské práce je prozkoumat možnosti a výhody publikování statistických dat na Sémantickém webu v souladu s principy Linked Data (kapitola 3). Pro publikování propojených statistických dat již existuje několik ontologií, například slovník Public Spending. Tato ontologie je však velmi specifická (proprietární) a nevyužívá slovníku Data Cube Vocabulary, tudíž je možnost vícenásobného použití (za různými účely) takto definovaných dat poměrně omezená. Ontologie Public Spending byla využita v praktické části pro návrh nové datové struktury (s pracovním názvem *Public Spending Data Cube,* viz kapitola 4), která se opírá o Data Cube Vocabulary. Návrh demonstruje, jak lze tento slovník aplikovat na již existující strukturu, a tím pádem rozšířit možnosti využití dat, která jsou dle této struktury definována. Znalostní bázi ontologie Public Spending lze převést pomocí SPARQL dotazu na tuto novou datovou strukturu, což umožnuje data zpracovávat pomocí analytických nástrojů pro statistická data.

Součástí bakalářské práce je také teoretický úvod (kapitola 2), který umožňuje i nepříliš znalým čtenářům nahlédnout do problematiky publikování dat na Sémantickém webu. V teoretické části je popsán postup zveřejnění dat, od jejich zápisu, editace, až po publikování na Webu (kapitola 2.4). Tyto informace jsou doplněny o odkazy na související zdroje, které rozšiřují danou tematickou oblast. Práce tedy může posloužit jako úvod do oblasti Linked Data, což může být užitečné pro čtenáře, kteří shání informace o této problematice v českém jazyce (téměř veškerá dokumentace či podrobnější informace jsou dostupné pouze v anglickém jazyce).

2 Sémantický web a principy Linked Data

2.1 Sémantický web

Sémantický web je rozšířením World Wide Webu, které doplňuje klasický „*web dokumentů*“ o možnost popisu dat, aby byla čitelná a pochopitelná nejen pro člověka, ale i pro stroje, to umožňuje data publikovaná na Webu automaticky zpracovávat. Za vývojem Sémantického webu stojí konsorcium *W3C*[1] zabývající se vývojem standardů pro World Wide Web. Sémantický web je podle W3C definován takto: *„Sémantický web je reprezentace dat na WWW.*[2] *Je to společné úsilí vedené W3C s velkým počtem účastníků z výzkumných a průmyslových partnerů. SW*[3] *je založen na Resource Description Framework (RDF), který integruje různé aplikace za použití XML*[4] *pro syntax a URI*[5] *pro pojmenování.“* ([27], str. 86).

Ačkoliv WWW obsahuje nepřeberné množství informací a dat, je velice obtížné s nimi efektivně pracovat, protože se vyskytují převážně v nestrukturované formě. Dobrá strukturovanost dat je klíčovým faktorem pro jejich využívání a pro vytváření nástrojů, které s těmito daty pracují. Z důvodu efektivního využívání publikovaných dat byly vyvíjeny různé přístupy k jejich strukturalizaci [17]. Mezi ně patří například *mikroformáty*, založené na přidávání významu a struktury přímo do HTML stránek. Mezi často používané mikroformáty patří například *hCard*[6] pro publikování informací o lidech a organizacích na Webu, nebo *hCalendar*[7] pro zveřejňování událostí. Značky dodávající textu význam se vkládají přímo do kódu stránky. Dalším z přístupů jsou tzv. „*Web API*“, které poskytují dotazovací přístup ke strukturovaným datům skrze protokol HTTP.[8] Tyto technologie jsou využívány i velkými společnostmi, jako je například *Amazon*[9] nebo *Flickr*.[10] Oba uvedené přístupy mají však zásadní nevýhodu. První zmíněná technologie je omezena malým

[1] World Wide Web Consorcium, více na http://www.w3.org/

[2] World Wide Web

[3] Zkratka „SW“ v tomto kontextu znamená „Sémantický web“

[4] EXtensible Markup Language – jazyk pro strukturovaný popis dat

[5] Uniform Resource Identifier – řetězec sloužící pro jednoznačný popis zdroje informací

[6] Specifikace hCard dostupná na http://microformats.org/wiki/hcard

[7] Specifikace hCalendar dostupná na http://microformats.org/wiki/hcalendar

[8] HyperText Transfer Protocol – protokol pro sdílení hypertextových dokumentů

[9] http://www.amazon.com/

[10] https://www.flickr.com/

množstvím typů používaných entit a atributů. Při použití Web API vznikají oddělené soubory dat, které nejsou navzájem propojeny, je tedy složité je vyhledat a využít je jinak, než bylo původně zamýšleno. Právě znovuvyužití dat je jedním z cílů Sémantického webu, především oblasti nazvané *Linked Data* (propojená data), kterým je věnována kapitola č. 2.2. Snahou komunity okolo propojených dat je propojení různých informačních zdrojů do jednoho globálního prostoru dat [17], ve kterém bude snadné data vyhledávat – například při vyhledání určité informace bude snadné dohledat i informace související.

Důležitým a v poslední době rozmáhajícím se trendem je publikování dat o veřejné správě, například o financování projektů nebo o veřejných zakázkách. Za tímto trendem stojí jak tlak veřejnosti na transparentní vedení státu, tak snaha samotného vedení tyto informace sdělovat (nejpokročilejšími zeměmi jsou v tomto směru Velká Británie a USA) [27]. Kromě transparentnosti s sebou otevřené publikování dat nese i další výhody. Takto publikovaná data lze různě analyzovat či vytvářet vizualizace pro jejich lepší pochopení. Tímto se konkrétně v oblasti veřejných výdajů zabývá například projekt Public Spending (viz kapitola 3.2), který shromažďuje data o vládních veřejných výdajích, a následně je analyzuje a propojuje [33].

2.1.1 Vlastnosti Sémantického webu

Existuje několik zásadních vlastností a předpokladů, které je potřeba brát v úvahu při práci s daty na Sémantickém webu. Všechny vychází ze základního předpokladu „*otevřeného světa*“ popsaného níže, který na rozdíl od „*předpokladu uzavřeného světa*“ (aplikovaného například v relačních databázích) umožňuje volněji šířit a publikovat informace. Tyto charakteristiky jsou důležité pro správný chod aplikací pracujících nad Sémantickým webem.

The Open World Assumption

Na rozdíl od publikování dat v databázích, u nichž se předpokládá, že jsou kompletní (a tudíž data, která nejsou v databázi obsažena, nejsou brána v úvahu), na Sémantickém webu tento předpoklad neplatí. Vždy je potřeba počítat s tím, že existují další relevantní informace z dosud neznámých zdrojů. Této vlastnosti se říká „*Předpoklad otevřeného světa*“. Informace o světě tedy nikdy nejsou úplné a kdykoliv se může objevit nová [3]. Neplatnost určité informace je tedy třeba vyjádřit explicitně [27].

Anyone Can Say Anything About Any Topic

Jelikož je RDF framework založený na principu otevřeného světa, *umožňuje komukoliv publikovat cokoliv o čemkoliv*. Každý tedy může zveřejňovat data o určité entitě a kombinovat s daty z jiných zdrojů [3]. Nevýhodou však je, že nelze nikomu zabránit, aby zveřejňoval nesmyslná nebo nekonzistentní data. Není zde totiž žádná autorita, která

kontroluje správnost informací. Na druhou stranu však tato vlastnost umožňuje vyjádření rozdílných názorů o určitém tématu. Je pouze na aplikaci/osobě, který zdroj bude považovat za důvěryhodný. Toto platí analogicky i na webu dokumentů, kde kdokoliv může zveřejnit webovou stránku s libovolnými údaji, a je pouze na uživateli, zda z této stránky bude čerpat [3].

The Network Effect

Pokud je lidem poskytnuta možnost něco vyjádřit (což je umožněno výše zmíněnou charakteristikou *„Anyone can say Anything about Any topic“*), jsou ochotni sami vytvářet nový obsah. S rostoucím množstvím dat se zvyšuje i použitelnost technologie, což přitahuje další uživatele. Tato vlastnost bývá nazývána *„The Network Effect“* a stojí i za rozšířením Sémantického webu.

The Non-unique Naming Assumption

Jedním z důležitých předpokladů, který se nazývá *„předpoklad nejedinečného pojmenování“*, je, že lidé mohou pro jednu entitu používat různé názvy. Z toho vyplývá, že odlišné URI mohou popisovat stejný objekt [3]. V takovém případě je u něj vhodné explicitně uvést, že na něj odkazují různé identifikátory.

The Data Wilderness

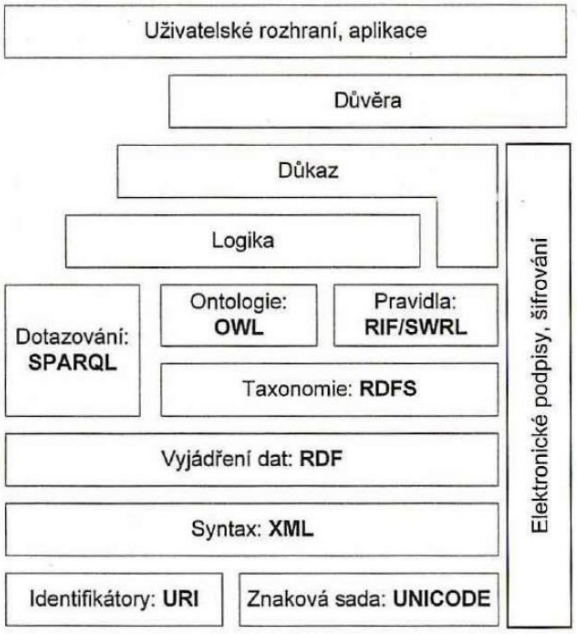
Takzvaná „*divokost dat*“ znamená, že Web sice může obsahovat cenné informace, ty však nemusí být snadno interpretovatelné. Mohou mít různou formu nebo jsou nekonzistentně organizovány [3].

2.1.2 Architektura Sémantického webu

K zajištění komunikace mezi uživateli Sémantického webu je potřeba dodržovat jednotnou syntaxi i vyjádření sémantiky [27]. Architektura Sémantického webu sestává z různých technologií a lze ji rozdělit do několika vrstev, jak znázorňuje ***Obrázek 1***. Ve spodních vrstvách jsou zobrazeny technologie použité pro syntax. Pro zápis dat byla zvolena znaková sada Unicode, především pro svou univerzálnost. Přiřazuje totiž libovolnému znaku unikátní kód nezávisle na jazyce nebo platformě [42]. Zdroje, ze kterých je Sémantický web tvořen, jsou definovány pomocí *URI* (Uniform Resource Identifier). URI jsou krátké řetězce, které identifikují zdroje na Webu. Díky tomu je snadné ke zdrojům přistupovat přes protokoly HTTP nebo FTP.[11] Použití URI je výhodné ze dvou důvodů. Poskytují jednoduchý způsob, jak vytvořit globálně unikátní názvy, což umožňuje decentralizaci uložených informací. Druhým důvodem je fakt, že URI může sloužit nejen

[11] File Transfer Protocol – protokol pro přenos souborů pomocí počítačové sítě

jako pojmenování, ale i jako přístup k informacím popisujícím daný objekt [17]. Každý záznam je poté představován trojicí prvků, která bývá nazývána *RDF trojice*. Každé části RDF trojice je tedy přiřazen jednoznačný název URI, výjimku tvoří tzv. „literály“[12] – data, na která je zbytečné odkazovat (například věk osoby nebo jiný číselný údaj). Aby nebylo nutné ke každému objektu udávat URI zdroje, byly zavedeny takzvané prefixy. URI zdroje je uvedena na začátku dokumentu spolu s řetězcem znaků, který je pak používán podle potřeby uvnitř dokumentu, čímž je zvýšena čitelnost textu a zkrácena délka dokumentu (nevznikají multiplikace).



Obrázek 1:
Architektura Sémantického webu ([27], s. 87)

RDF data bývají vyjádřena pomocí jazyků XML a XML Schema. Často využívanou syntaxí pro zápis RDF dat je *Turtle*,[13] která bývá považována za přehlednější a pro člověka čitelnější než normativní formát RDF/XML [27]. Pro expresivnější vyjádření informací jsou přidány jazyky RDFS a OWL, umožňující tvorbu taxonomií i složitějších ontologických konstruktů. Oba tyto jazyky jsou též vyjádřeny v RDF. Nad nimi lze provádět dotazování pomocí jazyka SPARQL. Součástí architektury jsou také jazyky RIF[14]

[12] Literál je konstanta, která je zapsána svoji vlastní hodnotou (například 9 je celočíselný literál, „slovo“ je řetězcový literál, apod.)

[13] Tense RDF Triple Language, specifikace dostupná na http://www.w3.org/TR/2012/WD-turtle-20120710/

[14] Rule Interchange Format – XML jazyk pro vyjádření pravidel, které mohou provádět počítače [42].

a SWRL,[15] které slouží k vyjádření pravidlových znalostí. Vrstva nazvaná „Logika“ představuje deskripční logiku, která je základem pro jazyk OWL. Při práci s daty na Sémantickém webu je podstatné, aby každý uživatel dokázal ve stejném čase odvodit stejné důsledky a zároveň měl důvěru v používaná data [27]. Důvěru zajišťují prostředky běžně používané na WWW – elektronické podpisy a šifrování.

2.2 Linked Data

Pojem Linked Data vyjadřuje soubor technologií a postupů, určujících, jak publikovat data na Sémantickém webu. Podle [5] jsou informace publikované na Webu dle principů Linked Data zveřejněny takovým způsobem, že jsou strojově čitelné, jejich význam je přesně definován, odkazují na externí zdroje a je zároveň možné na ně z těchto zdrojů odkazovat. Použití principů Linked Data při publikování informací na WWW umožňuje zajistit vyšší užitečnost těchto informací – taková data jsou jednoduše vyhledatelná a je možné je znovu využít. Základem velkého množství aplikací na Sémantickém webu, jako jsou sociální sítě, tezaury, kolaborativní publikování, a další, jsou ontologie,[16] na které poté odkazují různě distribuované dokumenty. Na rozdíl od toho aplikace Linked Data staví na rozměrných datových souborech, jimž poté přizpůsobují návrh ontologií [27]. Tato data jsou získávána různými způsoby, které jsou popsány v kapitole 2.4.2. Takto propojená data vytvářejí globální prostor dat (tzv. „Web of Data“), propojující informace o různých doménách – od informací o lidech, umění nebo společnostech, po statistické nebo vědecké údaje. Nad těmito informacemi pracují aplikace, které umožňují uživatelům prohlížet data z různých, avšak propojených datových zdrojů (generické Linked Data prohlížeče) nebo procházejí tento globální prostor skrze odkazy mezi zdroji a poskytují možnosti dotazování do „Webu dat“, jakoby to bylo jedno centrální úložiště (webové vyhledávače). Celá technologie staví na již existujícím World Wide Webu, nesnaží se ho nijak měnit, ale pouze doplňovat a rozšiřovat jeho schopnosti práce s daty.

2.2.1 Principy Linked Data

V roce 2006 přišel Tim Berners Lee s pravidly, jak propojená data publikovat. Protože se však Linked Data publikují na WWW, je třeba kromě těchto principů znát také architekturu klasického webu dokumentů, a to především technologie URI, protokol HTTP a jazyk HTML. Bez nich by nešlo na WWW zveřejňovat informace, bez znalosti principů

[15] Semantic Web Rule Language rozšiřující OWL o Hornovy klauzule

[16] Pojem *ontologie* pochází z oblasti filozofie a lze vysvětlit jako věda zabývající se nejobenějšími otázkami jsoucna a bytí [27]. V kontextu této práce mají však pojem ontologie odlišný význam – jsou to množiny přesně definovaných tvrzení o určité části světa (obvykle nazývané „oblast zájmu“) [21].

Linked Data však publikovat lze, ale ve formě, která nelze propojit [4].Tato pravidla jsou čtyři a jsou známé pod názvem „*Linked Data principles*“[17] [17]. Následující podkapitoly se těmto principům věnují.

Pojmenování objektu pomocí URI

První z principů doporučuje použití URI k jednoznačnému určení objektů. Na webu dokumentů byly za tyto objekty považovány pouze jednotlivé dokumenty, v rámci Linked Data se však doporučuje identifikovat nejen digitální obsah, ale i reálné a abstraktní objekty [17]. Jako příklad reálného objektu lze uvést konkrétní osobu, město nebo společnost. Jako abstraktní lze považovat určitou barvu nebo vlastnost objektu. Právě technologie unikátních identifikátorů zdrojů (URI) umožňuje identifikovat téměř cokoliv, a to jednoznačně napříč celým Webem. Je tedy třeba upustit od zavedené myšlenky, že URI slouží pouze k označení umístění nějakého zdroje, ale přijmout fakt, že ho lze použít jako jednoznačný název kteréhokoliv objektu.

URI vyhledatelné pomocí HTTP

Pro všechny identifikátory URI (jak webových dokumentů, tak reálných objektů) by mělo platit, že jsou vyhledatelné protokolem HTTP a že při zaslání požadavku klientem server vrátí tomuto klientovi popis zdroje, který identifikují. Tento popis je odeslán ve formě, jakou si žádá uživatel (požadavek na formát odpovědi je uveden v HTTP hlavičce). Lidé obvykle vyžadují odpověď ve formátu HTML, na rozdíl od toho jsou pro stroje čitelné informace zapsané v jazyce RDF. Tomuto přístupu se říká „*content negotiation*“ [17].

Při pojmenování reálných objektů pomocí URI je třeba rozlišit, zda se jedná o název reálného objektu nebo dokumentu, který tento objekt popisuje. Existují dva způsoby, jak lze identifikovat reálné objekty. První způsob se nazývá „*303 URI*“ a spočívá v tom, že server místo samotného objektu odesílá klientovi odpověď s HTTP hlavičkou „*303 See other*“, která uživatele přesměruje z URI objektu na URI dokumentu. Tato metoda je využívána u velkých množin dat a je flexibilní, jelikož lze cíl odpovědi nakonfigurovat zvlášť [17]. Druhým způsobem je tzv. „*hash URI*“. V tomto případě URI dokumentu obsahuje speciální část oddělenou od URI dokumentu pomocí „#“. Tato část popisuje konkrétní objekt popsaný dokumentem. Existence „#“ v identifikátoru zároveň znamená, že klient nemá odpověď zpracovat přímo. Hash URI využívají především menší množiny dat, například slovníky. Vyskytuje se taky při využití RDFa.[18] Tato metoda zašle méně HTTP požadavků než 303 URI, ale klient získá jako odpověď celý dokument, tedy i popisy objektů, o které nežádal. Proto se tento způsob nehodí pro velké datasety[19] [17].

---

[17] Originální dokument dostupný na http://www.w3.org/DesignIssues/LinkedData.html

[18] RDF by Attribute – způsob publikování RDF dat pomocí přímého zakomponování do HTML dokumentu

[19] Pojem dataset se používá pro množinu dat, která je určitým způsobem vymezena.

Použití RDF k poskytnutí užitečných informací o objektu

Aby mohly různé aplikace využívat Linked Data, je třeba, aby tato data byla zapsána ve standardizovaném formátu. Konsorcium W3C, zabývající se webovými technologiemi, zvolilo za tímto účelem technologii RDF. RDF je datový model reprezentující informace pomocí orientovaného grafu. Pro vyjádření tohoto grafu je třeba zvolit některý ze serializačních formátů. Bližší popis technologie RDF a jejího vyjádření je rozebrán v kapitole 2.3.1. Častým problémem je, že ačkoliv mnoho autorů publikuje ontologie v těchto formátech, samotná data jsou uložena odděleně, například v komprimovaných archivech, takže k nim nelze jednoduše přistupovat [4].

Zahrnutí odkazů na další objekty

Posledním pravidlem publikování informací v souladu principy Linked Data je zahrnutí odkazů na další objekty. Tento princip je velice důležitý, protože je páteří celého globálního prostoru dat. Dataset bez odkazů na jiné množiny dat by byl pouze fragmentem informací, který by byl nevyhledatelný. Propojení dvou datasetů lze provést pomocí trojice subjekt-predikát-objekt, kde subjektem je URI odkaz jednoho datasetu a objektem odkaz vedoucí na jmenný prostor datasetu jiného. Odkazy vedou na popis zdroje této množiny dat a je možné k nim připojit další RDF trojice, pomocí kterých lze vyhledat další zdroje. Tyto odkazy mohou být dle [17] trojího typu:

- Odkazy představující vztahy datového zdroje se souvisejícími objekty z jiných datových zdrojů. Například objekt popisující určitou osobu může vést na zdroj popisující společnost, kde osoba pracuje.
- Odkazy identifikující stejné objekty pomocí různých zdrojů. Jelikož podle předpokladu „*Non-unique Naming Assumption*“ může být objekt popsán více způsoby (zdroji), je vhodné tyto zdroje propojit.
- Odkazy mířící z popisu objektu na termíny ve slovnících. Připojení termínů z často používaných slovníků umožňují aplikacím přiřadit objektům konkrétní význam.

2.2.2 Výhody propojených dat

V dnešní době je World Wide Web plný obrovského množství informací, které nemohou být plně využity, protože jsou uloženy v různých formátech nebo nestrukturované formě. Zároveň sílí tlak opakované využívání těchto dat a na informační potřebu, jak za účelem výzkumu nebo získání konkurenční výhody v obchodním světě. Linked Data jsou řešením problému, jak tato data ukládat a publikovat, aby bylo snazší k nim přistupovat. Standardizace publikování a uložení ve strojově čitelné formě vede k vývoji aplikací, jež s takovými daty dokáží efektivně pracovat. Na rozdíl od zavedených technologií (jako je ukládání do relačních databází nebo ve formátech kancelářských balíků), Linked Data

zjednodušují objevení a vzájemné propojení těchto informací [17]. Způsob publikování dat v souladu s principy Linked data poskytují uživatelům standardizovaný mechanismus přístupu k informacím, implementace RDF přináší jednotný datový model a použití slovníků vede k tomu, že data jsou jednotně popsána, což umožňuje aplikacím znát význam těchto informací.

2.2.3 Linked Open Data

V současné době je velkým trendem publikování propojených dat pod otevřenou licencí. Rozsah a rozšíření takto zveřejňovaných dat vystihuje tzv. The Linking Open Data Cloud diagram[20] (viz ***Obrázek 2***). Tento graf vzniká pomocí sběru metadat o datasetech zveřejňovaných komunitou Linking Open Data Community[21] a zobrazuje datasety publikované pomocí principů Linked Data a jejich vzájemné vztahy. Dále jsou zde barevně odlišené různé oblasti zájmů – od dat zveřejňovaných vládními organizacemi nebo sociálními sítěmi, po geografické údaje nebo média. Tento diagram je periodicky aktualizován, nejnovější verze pochází ze srpna 2014. Základem otevřených dat je projekt DBpedia,[22] který spojuje velkou většinu datových zdrojů. Tento projekt má za cíl extrahovat informace z Wikipedie, ukládat je do strukturované formy a připojovat k nim data z jiných zdrojů.

Zajímavou oblastí Linked Open Data jsou zejména data publikovaná veřejnou správou – Open Government Data (na diagramu jsou znázorněna světle modře a zabírají většinu jeho levé části). Právě na tuto oblast se zaměřuje praktická část bakalářské práce. Pojem „*Open Government*“ zaveden memorandem nazvaném „*The Memorandum on Transparency and Open Government*“, které podepsal americký prezident Obama krátce po jeho inauguraci v lednu 2009 [25]. Ideu Open Government ([25], s. 9) popisuje takto: *„The basic idea of Open Government is to establish a modern cooperation among politicians, public administration, industry and private citizens by enabling more transparency, democracy, participation and collaboration.“*[23] Základním předpokladem pro fungování této myšlenky je volný přístup k informacím a možnost je využívat – to umožňují právě Open Government Data.

Pod pojmem Open Government Data se skrývá trend zpřístupňování vládních či veřejných administrativních dat a obsahu, s co nejmenšími bariérami přístupu a s možností

---

[20] Klikací verze diagramu dostupná na http://lod-cloud.net/versions/2014-08-30/lod-cloud.svg

[21] Více na http://www.w3.org/wiki/SweoIG/TaskForces/CommunityProjects/LinkingOpenData

[22] http://dbpedia.org

[23] „Základní myšlenkou Open Government je zřízení moderní spolupráce mezi politiky, veřejnou správou, průmyslem a soukromými subjekty, a to za účelem umožnění lepší transparentnosti, demokracie, účasti a spolupráce.“

znovuvyužití. Tato data jsou uložena tak, aby byla čitelná jak pro lidi, tak pro stroje. Jsou určena jak pro širokou veřejnost, tak pro média, akademickou půdu, politiky nebo státní úředníky [25]. Na konferenci v Kalifornii v roce 2007 byly definovány základní principy pro publikování otevřených vládních dat, které byly rozšířeny v roce 2010 nadací Sunlight Foundation na následujících 10 principů [[46], [25]]:

1. Data by měla být kompletní, tedy zahrnovat vše o konkrétní doméně
2. Data by měla být uložena v co nejhrubší formě[24] a obsahovat původní informace
3. Data by měla být zveřejněna včas, aby jejich obsah byl aktuální (to se týká především dat, která rychle zastarávají).
4. K datům by mělo být snadné přistupovat.
5. Data by měla být uložena ve strojově čitelné formě
6. Každý by mít k datům stejně snadný přístup, a to bez registrace či poplatků.
7. Data by měla využívat otevřené standardy, tedy čtení dat by mělo být možné pomocí programů, které lze získat bez jakýchkoliv poplatků.
8. Data by měla být publikována pod otevřenou licencí, aby je každý mohl číst a využívat.
9. Data by měla být vyhledatelná v průběhu času. Pokud jsou již neaktuální, měla by být uložena v dostupných archivech.
10. Data by neměla být přístupná za poplatek, ačkoliv existují náklady na jejich sbírání a zveřejňování.

Mezi společnosti využívající otevřená data patří například Světová banka,[25] shromažďující a zpřístupňující data o vývoji různých zemí. Dalšími zajímavými zdroji jsou UNdata,[26] systém pro přístup k datům, která zveřejňuje *The United Nations Statistic Divisons* nebo data publikovaná neziskovou organizací REEEP[27] (*Renewable Energy and Energy Efficiency Partnership*), podporující trh s obnovitelnými zdroji energie v rozvojových zemích, za účelem snížení CO_2 emisí. Na otevřená data lze narazit i v médiích, podporuje je například The New York Times.[28] Co se týče zveřejňování otevřených vládních dat, mezi nejaktivnější státy patří USA, Austrálie, státy Skandinávie

[24] Za nejhrubší formu je v tomto kontextu považován takový stav dat, kdy jsou data publikována tak, jak byla primárně vytvořena. To znamená, že by data neměla být nijak modifikována.

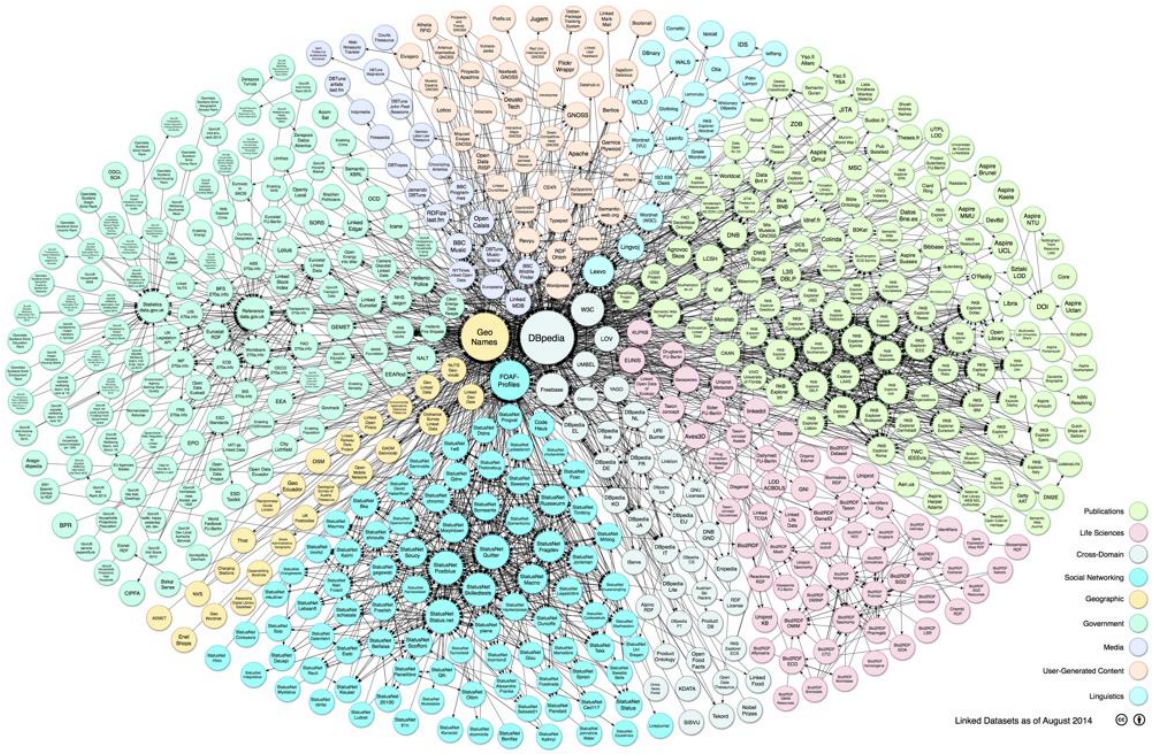
[25] Data Světové banky lze procházet na http://data.worldbank.org/

[26] http://data.un.org/

[27] http://data.reegle.info/

[28] Otevřená data publikovaná deníkem New York Times dostupná na http://data.nytimes.com/

a Velká Británie [25]. Publikováním otevřených dat v České republice se zabývá *Iniciativa za otevřenou datovou infrastrukturu*, kterou provozují akademičtí pracovníci a studenti z Fakulty informatiky a statistiky VŠE a Matematicko-fyzikální fakulty UK. Za zmínku stojí také projekt *Otevřená data*,[29] na jehož webové stránce lze najít různé informace o aktuálním dění kolem otevřených dat či zdroje, odkud čerpat data pro aplikace.



Obrázek 2:
Linking Open Data Cloud Diagram [11]

2.3 Reprezentace dat pomocí RDF a souvisejících technologií

Aby bylo možné data globálně propojit a přiřadit jim určitý význam, je zapotřebí použít standardizovanou technologii, která by toto umožnila. Jako standard pro reprezentaci dat o webových zdrojích byl zvolen datový formát Resource Description Framework. Ten byl původně navržen pro popis metadat, a hodí se tedy pro použití na Webu [27]. Samotné RDF je však pouze datový model, přiřazující zdrojům určité identifikátory, konkrétně se zde využívají identifikátory URI (Uniform Resource Identifier). Aby mohl být těmto identifikátorům přiřazen význam, je třeba zavést expresivnější technologii, nazývající se RDF Schema (RDFS). RDFS rozšiřuje RDF o sémantiku a využívá základ ontologií ve

[29] http://www.otevrenadata.cz/

formě taxonomie (tedy klasifikace třídící pojmy do slovníků, což například umožňuje jejich snadné vyhledávání). Pro složitější ontologické konstrukty byl vyvinut jazyk OWL (Web Ontology Language). Oba tyto jazyky jsou podrobněji popsány v kapitolách 2.3.2 a 2.3.3.

2.3.1 Resource Description Framework

RDF je rámec pro vyjádření informací o zdrojích (tedy o čemkoliv, od dokumentů na Webu, přes reálné objekty, až po abstraktní pojmy). RDF umožňuje, aby mohla být data zobrazena lidem, ale aby byla i strojově zpracovatelná. Používání RDF pro publikování informací s sebou nese i další výhody. Například je možné ve svých datasetech využívat data z jiných zdrojů, čímž je původní dataset obohacen o další informace. Do propojených datasetů je pak možné dotazování pomocí jazyka SPARQL. Lze propojovat například i informace o tom, že nějaká osoba zná jinou osobu, což vede ke vzniku distribuované sociální sítě [37].

RDF je datový model, jehož základem je popis dat pomocí orientovaného grafu [27]. Elementárním prvkem tohoto grafu, a zároveň představitelem jednoho tvrzení, je tzv. „RDF triple“ – trojice, která má strukturu <*subjekt*> <*predikát*> <*objekt*>. Subjektem a objektem mohou být libovolné související zdroje (URI nebo literály). Predikát („property“)[30] určuje povahu vztahu mezi subjektem a objektem a je vždy binární (to znamená, že je tento vztah vždy pouze mezi dvěma entitami). Jeden zdroj může v trojicích vystupovat jako subjekt nebo i jako objekt, což umožňuje definovat souvislosti mezi jednotlivými trojicemi. Graficky lze tyto trojice vyjádřit jako orientovaný graf, kde jednotlivé subjekty či objekty vystupují jako uzly a predikáty jako hrany orientované od subjektu k objektu. Aby bylo možné publikovat RDF data na Webu, je třeba je převést na některý ze serializačních formátů. Tyto formáty jsou mezi sebou převoditelné, takže popsání grafu jakýmkoliv serializačním formátem vede ke stejné množině RDF dat [37]. Normativním formátem pro zápis RDF dat je RDF/XML [27], nicméně velmi rozšířeným formátem je Turtle, využívaný i v této práci. Jednotlivé formáty jsou popsány v následujícím textu.

Serializační formáty RDF

Mezi serializační formáty se řadí RDF/XML, formáty rodiny Turtle nebo například RDFa. Jazyk Turtle je široce rozšířený serializační formát, a to z důvodu, že je čitelný pro člověka a dobře se v něm pracuje (je použit i při návrhu datové struktury v praktické části této práce). V Turtle je možné používat prefixy pro jmenné prostory (není tedy třeba vždy uvádět celý URI zdroje), nebo pokud jeden zdroj vystupuje jako subjekt ve více trojicích,

[30] vlastnost

lze subjekt uvést pouze jednou a k němu připojit vlastnosti a objekty oddělené středníkem. Ukázka dat zapsaných ve formátu Turtle je uvedena ve Výpis 1.

Výpis 1: *Ukázka dat zapsaných ve formátu Turtle*

```
@prefix :              <http://www.example.org/ontology#>

:Alice  rdf:type       :Person ;
        :knows         :Bob ;
        :owns          :Lisa ;
        :liveIn        :Prague .

:Lisa   rdf:type       :Cat .

:Prague rdf:type       :City .
```

Do rodiny jazyků Turtle mimo jiné patří také TriG, N-Triples a N-Quads. TriG rozšiřuje jazyk Turtle o možnost pojmenování jednotlivých popisovaných grafů. N-Triples umožňuje jednoduchý zápis RDF trojic. Jeden řádek zakončený tečkou představuje jednu trojici. Zdroje jsou definovány pomocí jejich URI umístěných ve špičatých závorkách, literály jsou ohraničeny anglickými uvozovkami. K literálům je také přiřazen datový typ (využívající standard XML Schema),[31] tak, že je za obsah literálu napojen pomocí symbolů „^^“ URI datového typu. K řetězcům je také možné připojit jazyk, ve kterém jsou napsány (například @en pro anglický jazyk nebo @cs pro jazyk český). N-Triples se často používá pro přenos velkého množství dat. Rozšířením *N-Triples* je *N-Quads*, umožňující doplnit RDF trojici o čtvrtý prvek, zachycující URI grafu, ve kterém je trojice obsažena [37]. Ukázka N-Triples je zobrazena v následujícím výpisu.

Výpis 2: *Ukázka dat zapsaných ve formátu N-Triples*

```
<http://www.example.org/ontology#Alice> <http://www.w3.org/1999/02/22-rdf-
syntax-ns#type> <http://www.example.org/ontology#Person>.
<http://www.example.org/ontology#Alice> <http://www.example.org/ontology#knows>
<http://www.example.org/ontology#> .
<http://www.example.org/ontology#Alice> <http://www.example.org/ontology#owns>
<http://www.example.org/ontology#Lisa> .
<http://www.example.org/ontology#Alice> <http://www.example.org/ontology#liveIn>
<http://www.example.org/ontology#Prague> .
```

[31] Specifikace dostupná na http://www.w3.org/XML/Schema

RDF/XML je původní formát pro zápis RDF dat. Pro popis RDF trojic využívá syntaxe XML. Tento formát je považován za normativní, většina aplikací by ho tedy měla podporovat [27]. Pro jeho nečitelnost je však určen spíše pro strojové zpracování, pro ruční editaci je vhodné dokument převést do čitelnějšího formátu, například do Turtle. Následuje ukázka dat zapsaných v tomto formátu (viz Výpis 3).

Výpis 3: *Ukázka dat zapsaných ve formátu RDF/XML*

```
<rdf:Description rdf:about="http://www.example.org/ontology#Alice">
    <rdf:type rdf:resource="http://www.example.org/ontology#Person" />
    <example:knows rdf:resource="http://www.example.org/ontology#Bob" />
    <example:owns rdf:resource="http://www.example.org/ontology#Lisa" />
    <example:liveIn rdf:resource="http://www.example.org/ontology#Prague" />
</rdf:Description>
```

Formát JSON-LD umožňuje RDF data zapisovat pomocí JSON syntaxe a používá se pro převod dat uložených v JSON do RDF (viz Výpis 4). Formát RDFa, sloužící pro zápis RDF dat přímo do HTML dokumentů je blíže popsán v kapitole 2.4.2.

Výpis 4: *Ukázka dat uložených ve formátu JSON-LD*

```
{
    "@context": {
                "@id":     "http://www.example.org/ontology#Alice",
                "@type":   " http://www.example.org/ontology#Person ",
                "knows":   "http://www.example.org/ontology#Bob",
                "owns":    "http://www.example.org/ontology#Lisa",
                "liveIn"   " http://www.example.org/ontology#Prague
}
}
```

2.3.2 RDF Schema

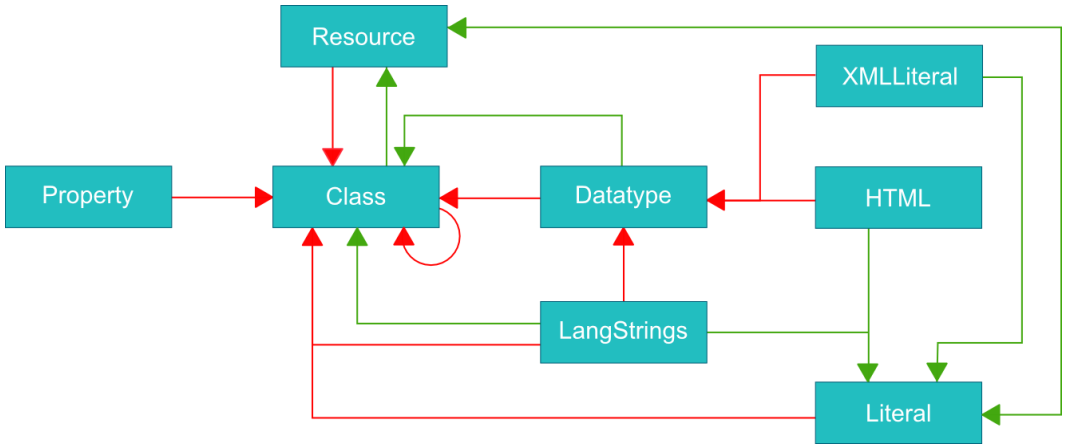
Samotný RDF model nevytváří předpoklad pro definování významu jednotlivých zdrojů. K vyjádření významu zdrojů tak, aby byl rozlišitelný i pro stroje, se využívá tzv. slovníků, které zavádí rozšíření RDF nazvané *RDF Schema* (dále pouze RDFS) [8]. RDFS do modelu RDF zavádí taxonomie, které jsou základem pro vytváření ontologií. V RDFS je tedy možné vyjádřit hierarchické vztahy mezi jednotlivými pojmy. RDF Schema je zapsáno pomocí RDF (vše je popsáno pomocí RDF trojic) a obsahuje definice tříd

a vlastností,[32] které lze použít při návrhu vlastních ontologií. Kromě vlastnosti *rdf:type* existující přímo v RDF a určující, že subjekt je instancí objektu, lze v RDFS také vyjádřit, že nějaký koncept (vlastnost) je podtřídou (podvlastností) jiného konceptu (vlastnosti). Systém tříd a vlastností definovaný v RDFS je velmi podobný systému používanému v objektově orientovaných jazycích. Na rozdíl od něj však definuje vlastnosti mimo třídy, jsou tedy prakticky nezávislé a je velmi snadné doplnit nové. Vztahy mezi vlastnostmi a třídami lze pak definovat pomocí klíčových slov *domain* and *range*, které slouží k odvození typu subjektu či objektu, který je použit ve vztahu s danou vlastností.

RDFS se tedy skládá z tříd a vlastností. Pojem „třída" („*class*") představuje skupiny zdrojů. Zdroje patřící do nějaké třídy se nazývají instance, tento vztah lze definovat pomocí *rdf:type*. RDFS umožňuje, aby třída mohla být instancí jiné třídy. Lze také definovat, že třída je podtřídou jiné třídy, a to pomocí vlastnosti *rdfs:subClassOf*. V případě, že určitá třída je podtřídou jiné třídy, instance této podtřídy jsou zároveň instancemi třídy nadřazené. V RDFS je definováno několik hlavních tříd, odrážející tuto důležitou vlastnost RDFS [8], které jsou znázorněny na ***Obrázku 3*** a popsány v následujících bodech:

- Všechny zdroje, které vystupují jako třídy, jsou instancemi třídy *rdfs:Class*. Samotná *rdfs:Class* je instancí sebe sama, protože je třídou.
- Všechny objekty popsané pomocí identifikátoru URI se nazývají zdroje. Veškeré zdroje jsou instancemi třídy *rdfs:Resource*. Všechny třídy (včetně *rdfs:Class*) jsou podtřídami *rdfs:Resource* a protože je *rdfs:Resource* třídou, je instancí třídy *rdfs:Class*.
- Všechny literály jsou instancemi třídy rdfs:Literal. Samotná tato třída je instancí třídy *rdfs:Class* a zároveň podtřídou rdfs:Resource.
- Třída *rdfs:Datatype* je třídou všech datových zdrojů. Je podtřídou a instancí *rdfs:Class* a každá instance této třídy je potřídou *rdfs:Literal*.
- Třída rdf:langStrings je podtřídou rdfs:Literal a instancí *rdfs:Datatype*. Je to třída všech řetězců, u kterých je definován jazyk.
- *rdf:HTML* je třída všech HTML hodnot v RDF dokumentu. Je instancí *rdfs:Datatype* a podtřídou *rdfs:Literal*. třída *rdf:XMLLiteral* je třída všech XML hodnot v RDF dokumentu a je též instancí *rdfs:Datatype* a potřídou *rdfs:Literal*.
- *rdf:Property* je třída všech vlastností definovaných v RDF. Je podtřídou *rdfs:Class*.

[32] Seznam tříd a vlastností definovaných v RDFS lze najít na http://www.w3.org/TR/rdf-schema/#ch_summary



Obrázek 3:
Vztahy mezi třídami definovanými v RDFS [8]

Druhým typem objektů definovaných v RDFS je „*vlastnost*“ (property). Vlastnosti představují vztah mezi zdrojem definovaným jako subjekt RDF trojice a zdrojem, který představuje objekt. Jednotlivé vlastnosti lze hierarchicky setřídit – to znamená, že u vlastnosti lze definovat, že je podvlastností jiné vlastnosti (pomocí *rdfs:subPropertyOf*). Právě vlastnosti *rdfs:subClassOf* a *rdfs:subPropertyOf* tvoří základní kámen vytváření taxonomií pomocí RDF, a tím pádem i základ pro návrh ontologií. Mezi další definované vlastnosti patří výše zmíněná rdf:type, a dále například *rdfs:label* pro vyjádření názvu zdroje čitelného pro lidi a *rdfs:comment* pro popis čitelný pro lidi. U každé vlastnosti lze odvodit, které prvky se při použití vlastnosti mohou objevit na místě subjekt a objektu. Pro subjekt je to vlastnost *rdfs:domain* (neboli definiční obor vlastnosti), pro objekt se používá *rdfs:range* (obor hodnot). Definiční obor a obor hodnot je specifikován i u výše zmíněných vlastností a je znázorněn v ***Tabulka 1***.

Tabulka 1: *Definiční obory a obory hodnot vybraných vlastností definovaných v RDF Schema [8]*

Název vlastnosti	Definiční obor	Obor hodnot
rdf:type	rdfs:Resource	rdfs:Class
rdfs:comment	rdfs:Resource	rdfs:Literal
rdfs:domain	rdfs:Property	rdfs:Class
rdfs:label	rdfs:Resource	rdfs:Literal
rdfs:range	rdfs:Property	rdfs:Class
rdfs:subClassOf	rdfs:Class	rdfs:Class
rdfs:subPropertyOf	rdfs:Property	rdfs:Property

Použití RDFS

Použití RDFS dodává návrhu ontologií celkem omezenou sémantiku, avšak je hojně využíváno při návrhu jednodušších ontologií. Důležitou schopností RDF Schema je možnost odvozování nových faktů (inference). To je umožněno existencí odvozovacích pravidel a axiomů (formulí). Dále umožňuje vytvářet hierarchické vztahy tříd a vlastností, což samotné RDF neumožňuje. V neposlední řadě je třeba zmínit možnosti používání různých slovníků, které slouží pro přesný popis informací z různých oblastí. RDFS také umožňuje lepší výměnu informací mezi distribuovanými datasety, což je způsobeno schopností RDFS oddělit od sebe definici tříd a jejich instancí (individuí). Datasety poté mohou mezi sebou sdílet informace o tom, jaké třídy se v nich vyskytují a kolik jejich instancí obsahují [9].

2.3.3 Web Ontology Language

Ačkoliv RDF Schema zavádí taxonomické vztahy, pro modelování komplexnějších datových struktur mohou být tyto možnosti nedostačující. Proto byl vyvinut komplexní jazyk pro konstrukci ontologií na Webu, nazvaný Web Ontology Language (dále jen OWL). OWL je výpočetní jazyk založený na logice a slouží k reprezentaci rozsáhlých znalostí o entitách a vztazích mezi nimi. Je třeba zdůraznit, že se nejedná o programovací jazyk, nýbrž jazyk deklarativní, popisující situace za pomocí logiky. Dokumentům v jazyce OWL se říká *ontologie*[33] [21]. Jazyk také umožňuje odvozovat další fakta, která nejsou explicitně zapsána (k tomu slouží nástroje zvané reasonery), což je velice důležitá vlastnost tohoto jazyka. OWL je vysoce expresivní jazyk, ve kterém lze vyjádřit komplexní ontologie, avšak na úkor výpočetní složitosti. Je proto důležité věnovat pozornost tomu, zda výpočetní složitost algoritmu použitého pro zpracování ontologie nepřesahuje hranici, kdy je daná ontologie nepoužitelná.

OWL, na rozdíl od RDFS, poskytuje mnohem širší slovník pro popis entit. Kromě prvků popsaných v RDFS (některé jsou popsány více do detailu, například OWL rozlišuje datové a objektové vlastnosti) totiž obsahuje i možnost definice ekvivalence pojmů z různých datasetů (vlastnost *owl:sameAs*), nebo například vlastnosti pro restrikci (*owl:allValuesFrom*). Pro oblast propojených dat však v mnohých případech stačí schopnosti RDFS a OWL se v nich příliš nepoužívá. OWL bývá často uplatněn u složitějších ontologií, například v biomedicíně [34][27].

[34] Touto oblastí se zabývá například http://www.bioontology.org/

2.4 Data na Sémantickém webu

2.4.1 Editace dat pro Sémantický web

Existuje řada specializovaných nástrojů, jež umožňují a zjednodušují ruční vytváření ontologií a znalostních bází. Tyto nástroje lze dělit na textové a grafické. Textové editory slouží ke snazší práci s již definovanými koncepty či lepší orientaci ve zdrojovém kódu. Pro editaci komplexnější ontologií mohou posloužit grafické editory s uživatelským rozhraním, ve kterém lze pracovat i bez hlubší znalosti syntaxe serializačních formátů. Tyto editory také podporují automatické uvažování, důležité nejen pro kontrolu konzistence ontologie, ale také pro odvození dalších faktů [27]. Mezi nejvyužívanější grafické editory lze zařadit Protégé,[35] NeOn Toolkit,[36] Neologism[37] nebo komerční TopBraid[38] [44]. Zástupcem jednodušších textových editorů je například XTurtle.[39] Editory XTurtle a Protégé byly využity pro praktickou část této práce a jejich funkcionalita je dále popsána v následujících podkapitolách.

XTurtle

XTurtle je textový editor dostupný jako plugin do vývojového prostředí Eclipse. Slouží pro editaci souborů psaných v Turtle syntaxi. Program podporuje zvýraznění syntaxe, navigaci, automatické doplňování kódu nebo jeho skládání do jednotlivých skupin [2]. XTurtle je vhodné použít pro editaci méně komplexních ontologií nebo znalostních bází, nehodí se však pro rozsáhlé RDF soubory. Nástroj nepodporuje dynamické načítání zdrojů z Webu, pro správný běh aplikace je tedy třeba, aby všechny používané slovníky byly umístěny jako soubory přímo na úložišti. Často používané slovníky lze stáhnout v balíčku[40] připraveném vývojáři programu.

---

[35] Program a dokumentace dostupná na http://protege.stanford.edu/

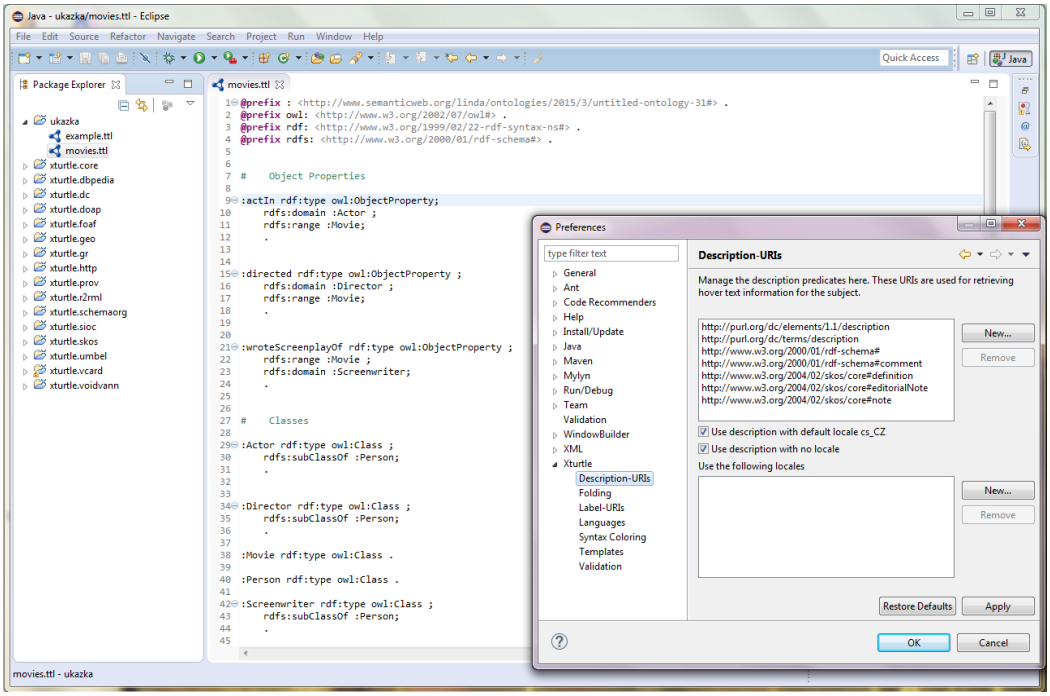
[36] Program a dokumentace dostupná na http://neon-toolkit.org/

[37] http://neologism.deri.ie/

[38] http://www.topquadrant.com/downloads/topbraid-composer-install/

[39] http://aksw.org/Projects/Xturtle.html

[40] Soubor s výchozími balíčky lze stáhnout na https://github.com/AKSW/Xturtle/wiki/Installation#importing-default-vocabulary-into-the-workspace



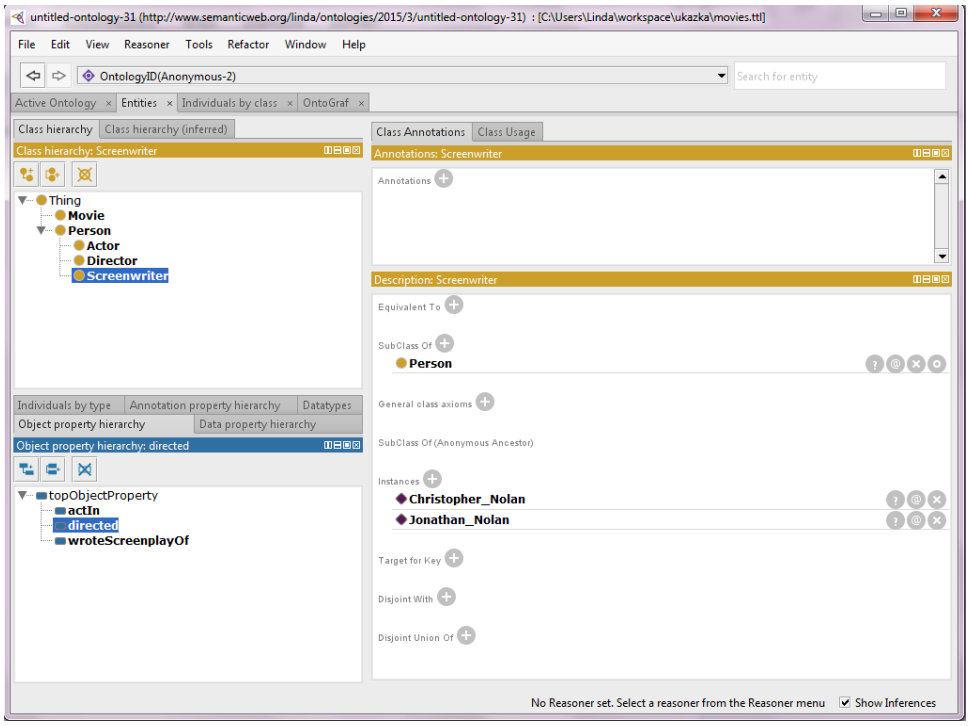
Obrázek 4:
Uživatelské rozhraní programu XTurtle

Na ***Obrázku 4*** lze vidět uživatelské prostředí programu XTurtle. V levé části se nachází seznam projektů (v tomto případě i s importovanými slovníky). Vpravo od levého panelu se nachází samotný textový editor s ukázkovým příkladem. XTurtle zvýrazňuje syntaxi Turtle, barevně odlišuje základní klíčová slova, komentáře a další prvky zdrojového kódu. Další funkcí nástroje je seskupování částí kódu do jednotlivých oddílů, které je možné pomocí tlačítka skrýt či rozbalit. Po instalaci pluginu v nastavení přibyla záložka, kde je možné chování programu nastavit (od customizace barevného zvýraznění syntaxe, až po způsoby validace zdrojového kódu).

Protégé

Oproti výše zmíněnému nástroji Protégé disponuje širší funkcionalitou a je vhodné ho používat při vytváření či editaci složitějších ontologií. Program je vyvíjen na Stanfordově univerzitě a ačkoliv byl původně zamýšlen pro tvorbu ontologií a znalostních bází založených na rámcích, je dnes uzpůsoben editaci ontologií v jazyce OWL. Kromě základních nástrojů lze do programu doinstalovat řadu pluginů,[41] rozšiřující funkcionalitu například o vizualizaci ontologie, automatické uvažování a další [27].

[41] Seznam pluginů je dostupný na http://protegewiki.stanford.edu/wiki/Protege_Plugin_Library



Obrázek 5:
Uživatelské rozhraní programu Protégé

Uživatel programu pracuje v jednotlivých záložkách. Při otevření či vytvoření nového projektu je aktivní záložka obsahující základní informace o ontologii – lze zde doplnit anotace, importovat používané slovníky nebo definovat URI, na kterém bude projekt dostupný online. Na ***Obrázku 5*** je znázorněna záložka pro úpravy jednotlivých konceptů a vlastností. V levém panelu se nachází hierarchie tříd, objektových a datových vlastností a také anotací nebo instancí tříd. V pravé části okna se po výběru prvku z levého panelu objeví možnosti, které lze tomuto prvku přiřadit. Protégé umožňuje celý projekt zobrazit i graficky, a to například pomocí pluginů OntoGraph či OWLViz[42]. Také je zde možné automaticky odvodit další vztahy mezi entitami, vyplývající ze zadaných vlastností. K tomu slouží takzvané reasonery,[43] kterých existuje celá řada.

2.4.2 Publikování propojených dat

Aby bylo možné publikovat informace jako Linked Data, musí tyto informace splňovat alespoň minimální požadavky na jejich formu a obsah. V prvé řadě musí být data identifikována dereferencovaným URI. Pokud je na taková data zaslán požadavek, který

[42] Pro vizualizaci ontologií existuje celá řada nástrojů, pro výběr vhodného vizualizačního nástroje lze využít Ontology Visualization Tools Recommender, dostupný na http://owl.vse.cz:8080/OVTR/recommendations.jsp

[43] Reasonery pro OWL lze najít na http://www.w3.org/2001/sw/wiki/OWL/Implementations

má nastavený MIME typ jako „application/rdf+xml“, datový zdroj by měl vrátit popis tohoto zdroje ve formátu RDF/XML. Dalším požadavkem je, že pokud URI identifikuje ne-informativní zdroj, musí datový zdroj vracet HTTP odpověď obsahující zprávu *„HTTP 303 Redirect“* o přesměrování na informační zdroj o tomto zdroji (viz kapitola 2.2.1). Alternativním řešením je připojení identifikačního fragmentu k URI, který popisuje související informační zdroj. Hlavním předpokladem propojených dat je provázanost informací do jedné globální sítě, což není zaručeno pouhým zveřejněním na Web. Aby daná data nebyla odděleným fragmentem informací, je třeba se ujistit, že na dataset odkazují některé externí zdroje. To způsobí, že jsou data nalezitelná a zpracovatelná vyhledávači [6]. Jedním ze způsobů, jak propojit dataset do globální sítě dat, je odkázat na něj na autorově FOAF[44] profilu, který je propojen s dalšími pomocí vlastnosti *foaf:knows*. Druhou možností je přesvědčit autory souvisejících datových zdrojů, aby automaticky vygenerovali RDF odkazy na URI daného datasetu, případně je vygenerovat a zaslat autorům, například komunitě DBpedia,[45] která často slouží jako pojítko pro propojování různých datových zdrojů [6]. Existuje několik způsobů, jak Linked Data publikovat, tyto způsoby jsou graficky znázorněny na ***Obrázku 6***. Aby v dané situaci byl zvolen nejvhodnější způsob, je třeba si podle [6] nejprve odpovědět na několik otázek:

Jak velký objem dat bude publikován?

Rozsah datového souboru je velmi rozhodující pro výběr způsobu zveřejnění na WWW. Ke zveřejnění RDF trojic v rozsahu do několika set trojic je možné zvolit statické RDF soubory [6]. Pro větší objem dat se doporučuje použít tzv. „*triplestores*“ – databázové systémy konstruované za účelem ukládání RDF trojic.

Jakým způsobem jsou původní data uložena?

Pokud jsou data uložena v relačních databázích, je vhodné použít platformu zvanou D2R Server pro publikování relačních dat na Sémantickém webu. Pokud je množina dat dostupná skrze Web API, je možné využít tzv. wrappery [6]. Trojice uložené v RDF databázích či v jednotlivých souborech lze publikovat přes související rozhraní nebo na webovém serveru.

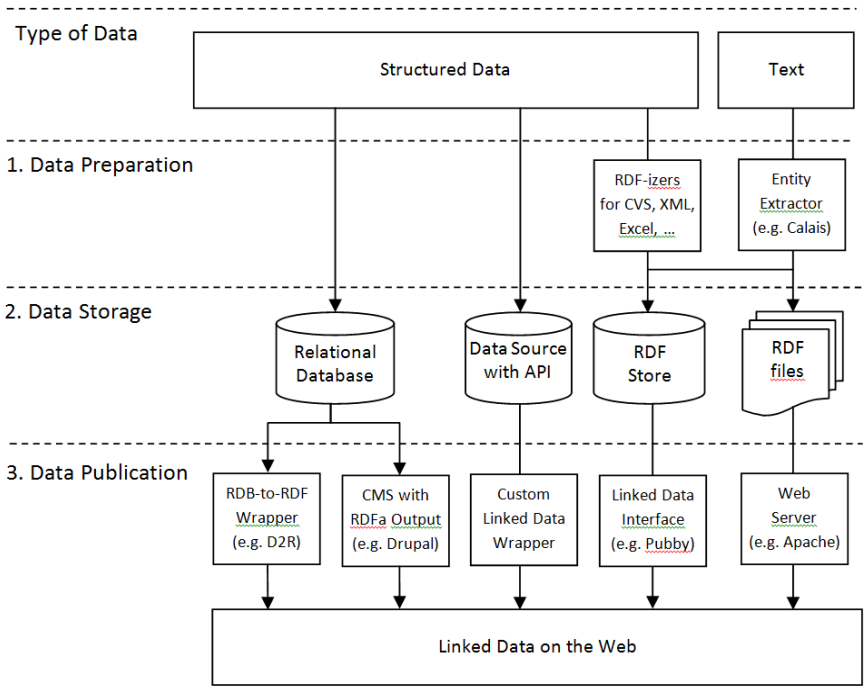
---

[44] FOAF (Friend Of a Friend) = ontologie popisující osoby a a vztahy s jinými objekty pomocí RDF. Specifikace dostupná na: http://xmlns.com/foaf/spec/

[45] http://dbpedia.org/

Jak často se mění obsah dat?

U informací uložených v jednoduchých RDF souborech může nastat problém v případě, mají-li často měnit svůj obsah, a to z důvodu, že tento způsob uložení dat nepodporuje pokročilejší práci s daty. V takovém případě je lepší využít komplexnější technologie, například D2R servery nebo triplestory.



Obrázek 6:
Způsoby publikování propojených dat [17]

SPARQL endpoint

SPARQL endpoint je služba protokolu SPARQL, umožňující uživatelům dotazování do množiny RDF dat [6], uložených různým způsobem (tyto způsoby jsou popsány v následujících podkapitolách). Analogicky lze SPARQL dotazování přirovnat k dotazování do relačních databází v jazyce SQL.[46] Protože je výsledek těchto dotazů uložen v různých strojových formátech, je SPARQL endpoint brán jako strojově přívětivé rozhraní pro znalostní báze [36]. Pojem „*endpoint*“ (v překladu koncový bod) [15] definuje jako *"A Web service endpoint is a (referenceable) entity, processor, or resource to which*

[46] Structured Query Language je dotazovací jazyk do relačních databází.

Web service messages can be addressed. Endpoint references convey the information needed to address a Web service endpoint."[47]

Pro konstrukci SPARQL dotazů je používán jazyk SPARQL. Syntax tohoto jazyka je podobná jazyku Turtle a struktura dotazu je podobná dotazům v jazyce SQL. Rozdílem je, že SPARQL umožňuje hledání RDF trojic [27]. Při vytváření dotazu se nejprve uvedou používané prefixy, následuje hlavička dotazu, skládající se z klauzule určující typ dotazu,[48] a proměnných, které se mají objevit ve výsledku dotazu. Klauzule SELECT je používána pro výpis RDF trojic z existujícího datasetu, na rozdíl od toho klauzule CONSTRUCT vytváří nový RDF graf [16]. Aby byly proměnné odlišeny, přidává se před jejich název znak „?“. Následuje tělo dotazu definované klauzulí WHERE a obsahující tzv. *vyhledávácí vzor*, vyjadřující RDF trojice, které jsou hledány [27]. Výsledek SPARQL dotazu (za použití klauzule SELECT) bývá zobrazen jako tabulka, jejíž sloupce odpovídají dotazovaným proměnným. Pro strojově čitelnou reprezentaci tohoto výsledku podporuje SPARQL endpoint různé formáty, například XML,[49] JSON,[50] CSV[51] nebo TSV[52] [41].

Statické RDF soubory

Nejjednodušším způsobem, jak publikovat RDF data na WWW, je zveřejňování statických RDF souborů. Tento způsob je vhodné zvolit, pokud je spravován poměrně malý objem informací [17]. Vytvořený RDF soubor se umístí na Web, například na webový server Apache pomocí FTP klienta. Tento soubor by měl mít správně nastaven typ MIME. Dále by mělo být zajištěno, že je dataset objevitelný. To lze provést přidáním tagu <link> do hlavičky HTML dokumentu popisujícího související zdroj, do nějž se uvede URI dokumentu v RDF. Tento způsob je nazýván „*autodiscovery pattern*“ [17].

RDFa

Další možností, jak zveřejňovat RDF data, je jejich zakomponování přímo do HTML stránek. K tomuto účelu slouží formát RDFa.[53] RDFa je často využíváno k publikování dat

---

[47] Definici lze přeložit jako „*Koncový bod webové služby je (odkazovatelná) entita, procesor nebo zdroj, na které mohou být zprávy webových služeb směřovány. Odkazy na koncový bod sdělují informace potřebné k oslovení koncového bodu webové služby*.“

[48] Mezi tyto klauzule se řadí SELECT, CONSTRUCT, INSERT, DESCRIBE a ASK [16]

[49] Extensible Markup Language, specifikace dostupná na http://www.w3.org/XML/

[50] JavaScript Object Notation, specifikace dostupná na https://www.ietf.org/rfc/rfc4627.txt

[51] Comma Separated Values, specifikace dostupná na https://www.ietf.org/rfc/rfc4180.txt

[52] Tab Separated Values, specifikace dostupná na http://www.iana.org/assignments/media-types/text/tab-separated-values

[53] RDF in Attributes, specifikace dostupná na http://www.w3.org/TR/xhtml-rdfa-primer/

v redakčních systémech CMS[54] jako je Wordpress nebo Drupal, u nichž je obsah publikován za pomoci již existujících šablon [17]. Velkou výhodou použití RDFa je, že data zapsaná pomocí této technologie jsou vyhledatelná i „běžnými“[55] vyhledávači, čímž obohacují vyhledávání o další informace.

RDF data se do dokumentu nevkládají ve formě komentářů, nýbrž je lze vkládat přímo pomocí značek, které jsou definovány v modelu HTML dokumentů zvaném DOM[56] [17]. Tyto značky lze přidat k HTML tagům jako atributy a tím k těmto tagům přidat význam definovaný pomocí URI. Důležité je, aby byly odlišeny objekty reálného světa, které daná data popisují, od dokumentů HTML+RDFa, které tyto objekty vyjadřují. K tomu slouží atribut „about“, který reálnému objektu přiřadí odkaz URI. Díky RDFa je možné dokumentu přiřadit výchozí jmenný prostor, a to za pomocí atributu „*vocab*“, jehož hodnotou je URI daného slovníku. V takovém případě není třeba do atributů vkládat celé URI, ale pouze klíčová slova ze slovníku. Pro vyjádření vlastnosti mezi subjektem a objektem slouží atribut „*property*“, pro vyjádření instance nějaké třídy lze použít atribut „*typeof*“ [20].

Pro práci s RDFa existuje velké množství užitečných nástrojů, mezi něž patří například RDFa Play,[57] editor dostupný online v prohlížeči. RDFa Play pracuje v reálném čase a při editaci kódu v levé části stránky se zároveň zobrazuje výsledný vzhled HTML dokumentu a také vizualizace sémantických prvků v dokumentu. Za zmínku také stojí nástroj Structured Data Linter,[58] který dokáže rozpoznat strukturované informace, které obsahuje stránka zadaná do vyhledávače tohoto nástroje. Jiné nástroje dovedou ze stránky extrahovat strukturované informace a uložit je do některého z RDF serializačních formátů – například do Turtle, N-triples nebo XML/RDF. Jako příklad lze uvést RDFa Distiller and Parser,[59] schopný ukládat výstup do všech uvedených formátů. Pro převod mezi těmito formáty pak lze využít například nástroj RDF Distiller.[60]

U programů pro extrakci strukturovaných dat může nastat problém, pokud se snaží vytěžit data z HTML stránek, které jsou vygenerované pomocí server-side skriptů (tyto skripty využívají právě některé redakční systémy, například Wordpress). Ty v některých případech mohou obsah stránek vypsat špatně a mohou vzniknout nekonzistence. Tento

---

[54] Content Management System

[55] Běžným vyhledávačem je v tomto textu myšlen internetový vyhledávač pro hledání webových dokumentů, jako je například Google.

[56] Document Object Model je objektový model definující strukturu HTML a XML dokumentů

[57] Dostupný nahttp://rdfa.info/play/

[58]Dostupný na http://linter.structured-data.org/

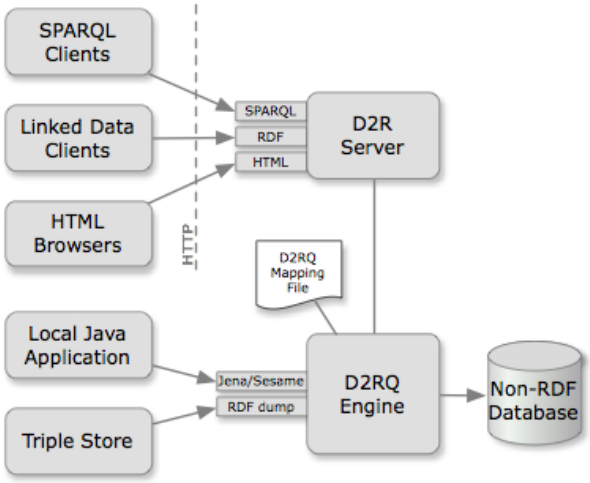
[59] Dostupný na http://www.w3.org/2012/pyRdfa/

[60] Dostupný na http://rdf.greggkellogg.net/distiller

problém lze vyřešit dvojím vytvořením obsahu ze stejné databáze, a to jednou do HTML a podruhé do RDF/XML. Pro generování RDF souborů pomocí server-side skriptů existují různé knihovny, například pro PHP lze uvést ARC2.[61] Následně je nutné odkázat na tento RDF soubor, například pomocí výše zmíněné *autodiscovery pattern*.

Publikování dat z relačních databází

Dalším způsobem pro publikování dat, která jsou původně uložena v relačních databázích, je publikování Linked Data view do relační databáze. View (pohled) bývá označován jako virtuální tabulka databáze, ale sám o sobě neuchovává žádná data, pouze definici tohoto pohledu [39]. Nástroje pro tento způsob publikace RDF dat se nazývají D2R servery. Jako konkrétní řešení lze uvést platformu D2RQ. Ta se skládá z D2RQ mapovacího jazyka pro popis vztahů mezi modelem relační databáze a dané ontologie, D2RQ enginu a D2R serveru, který poskytuje Linked Data view pro debuggování a SPARQL endpoint. Architekturu platformy D2RQ popisuje ***Obrázek 7.***



Obrázek 7:
Architektura D2R serveru [13]

RDF triplestores

V takzvaných „*triplestores*“ jsou data ukládána ve formě trojice <subjekt> <predikát> <objekt> [29]. Na rozdíl od relačních databází nepotřebují návrh komplexní datové struktury, protože jsou data jednoduše uložena jako RDF trojice. S tím je spojena i jednoduchost výsledku dotazů provedených do tohoto úložiště. SQL dotaz do relační databáze často vrací výsledek z různých tabulek, výsledek SPARQL dotazu do triplestore vrací jednoduchou tabulku s vyhledanými trojicemi [29].

[61] specifikace dostupná na https://github.com/semsol/arc2/wiki

Web API Wrappery

Velké společnosti, jako je Amazon, Twitter nebo Flickr ukládají svá data, určená k dalšímu použití, pomocí webových API. Taková data však nejsou uložena v jednotném formátu a využívají různě konstruované dotazy. Pro zapojení těchto dat do globální sítě lze používat tzv. *wrappery*. Ty zastávají funkci prostředníka mezi uživatelem a webovou API. Dle [17] nejprve zdroji dat přiřadí adresu URI, při požadavku na data z tohoto zdroje wrapper přepíše tento požadavek na požadavek konkrétní API a následně převede výsledek do RDF a zašle uživateli.

2.4.3 Práce s daty na Sémantickém webu

Jakmile jsou informace publikovány v souladu s principy Linked Data, stávají se strojově čitelná a vyhledatelná aplikacemi určenými pro jejich zpracování. Tyto aplikace mohou být rozděleny na dva typy – generické aplikace a specializované aplikace zaměřené na určitou oblast zájmu. Generické aplikace, určené pro práci s daty z jakékoliv oblasti lze dělit na vyhledávače a prohlížeče [17].

Vyhledávače

Linked data vyhledávače shromažďují tisíce informací z různých zdrojů. Neumožňují pouze získávat konkrétní informace, ale často zobrazují i jejich strukturu. Pro vyhledávání zdrojů pomocí klíčových slov lze použít například vyhledávač Watson[62] nebo LOD Cloud Cache.[63] Některé jiné vyhledávače, například <sameAs>,[64] umožňují k zadanému URI nalézt zdroje související [18].

Prohlížeče

Stejně jako „klasické“ webové prohlížeče dovolují uživateli procházet webové dokumenty skrze hyperlinky, prohlížeče určené pro Linked Data toto umožňují pomocí RDF odkazů [17]. To vede například k situaci, kdy lze jednoduše po vyhledání určitého regionu nalézt slavné osobnosti, které zde mají místo narození a dále seznam děl, které tyto osobnosti vydaly. Díky tomu lze tedy jednoduše procházet informace ze zdrojů, které nemusí být uloženy na jednom úložišti.

---

[62] http://watson.kmi.open.ac.uk/WatsonWUI/

[63] http://lod.openlinksw.com/

[64] http://sameas.org/

3 Analýza slovníků Sémantického webu pro statistická data

Velmi rozsáhlou oblastí propojených dat jsou různé statistiky. Statistická data totiž slouží k různým analýzám, testování různých vědeckých teorií, nebo například pro predikce v různých oblastech. A právě Sémantický web je ideální platformou pro publikování těchto informací – umožňuje shromažďovat data z heterogenních a distribuovaných zdrojů, aby s nimi šlo dále pracovat [10]. K tomuto účelu vznikl standard SDMX a slovník Data Cube Vocabulary, které umožňují publikovat a sdílet statistická data jako Linked data (obě technologie jsou popsány v následující kapitole). Tento slovník, jenž je neproprietární, je obecně doporučován pro návrh datových struktur pro statistická data a lze jej použít v různých oblastech. Vedle něj existují proprietární slovníky, které jsou úzce zaměřené na konkrétní doménu, jako příklad byl zvolen slovník Public Spending, blíže popsaný v kapitole 3.2.

Vedle slovníku Data Cube Vocabulary existují další ontologie pro práci se statistickými daty, například Statistics Ontology[65], která se též nazývá STATO. Umožňuje definovat procesy, jako jsou například statistické testy, jejich podmínky, apod. Ontologie také zahrnuje informace potřebné k vyvození závěrů ze statistických postupů, například pravděpodobnost, rozdělené, proměnné, rozptyl a další statistické prvky [22]. Pro popis statistických analýz lze využít také slovníku Linked Statistical Data Analysis.[66]

3.1 Data Cube Vocabulary

K reprezentaci statistických dat na Webu slouží Data Cube Vocabulary, využívající framework RDF a principy Linked Data. Tento slovník umožňuje popis dat pomocí více než dvou dimenzí – je postaven na takzvaném cube modelu, popisujícím statistický dataset jako multidimenzionální prostor (hyperkrychli indexovanou jednotlivými dimenzemi). Mezi hlavní přednosti Data Cube Vocabulary tedy patří multidimenzionalita a také možnosti vytvářet libovolné pohledy na data (k tomu slouží tzv. slices). Takto vytvořený model je poté kompatibilní se standardem pro publikování statistických dat jako Linked data. SDMX[67] (Statistical Data and Metadata Exchange) je ISO standard pro výměnu a sdílení statistických dat. SDMX mimo jiné poskytuje logický model pro popis

[65] více na http://stato-ontology.org/

[66] více na http://csarven.ca/linked-statistical-data-analysis, definice slovníku dostupná na http://stats.270a.info/vocab

[67] více na http://sdmx.org

statistických dat spolu s pokyny, jak tato data strukturovat. Důležitou součástí standardu je tzv. „*Content-oriented Guidelines*“,[68] který definuje množinu statistických pojmů a číselníků. Tyto pojmy jsou definovány pomocí RDF pro jednotné použití napříč statistickými daty na Webu. Data Cube Vocabulary staví kromě standardu SDMX na následujících šesti existujících slovnících [12].

- **DC (Dublin Core Terms)**[69] pro popis metadat;
- **FOAF (Friend Of a Friend)**[70] pro vyjádření informací o agentech;
- **ORG (The Organization Ontology)**[71] pro popis organizací;
- **SCOVO (The Statistical Core Vocabulary)**[72] pro definici statistických struktur;
- **SKOS (Simple Knowledge Organization System)**[73] pro schémata pojmů;
- **VoiD (Vocabulary of Interlinked Datasets)**[74] pro přístup k datům.

Výhodou Data Cube Vocabulary je to, že je velice obecný, lze jej použít pro popis dat z různých oblastí. Jeho úroveň opakovaného využívání je tedy na rozdíl od proprietárních slovníků velmi vysoká. Lze na něm stavět různé nové ontologie a data, která mají takto definovanou strukturu, lze dále zpracovávat, například pomocí analytických programů, popsaných v kapitole 3.1.4

3.1.1 Architektura Data Cube Vocabulary

Základem publikování statistických dat jako Linked Data za pomoci Data Cube Vocabulary je oddělení popisu datové struktury a samotných dat. Datová struktura je v podstatě popis datového modelu, pomocí něhož budou data publikována. Skládá se ze tří druhů komponent – dimenzí, měření a atributů. Samotná statistická jsou popsána pomocí

[68] dokumentace dostupná na http://sdmx.org/wp-content/uploads/2009/01/00_sdmx_content-oriented_guidelines_2009.pdf

[69] Specifikace dostupná na http://purl.org/dc/terms/

[70] Specifikace dostupná na http://xmlns.com/foaf/spec/

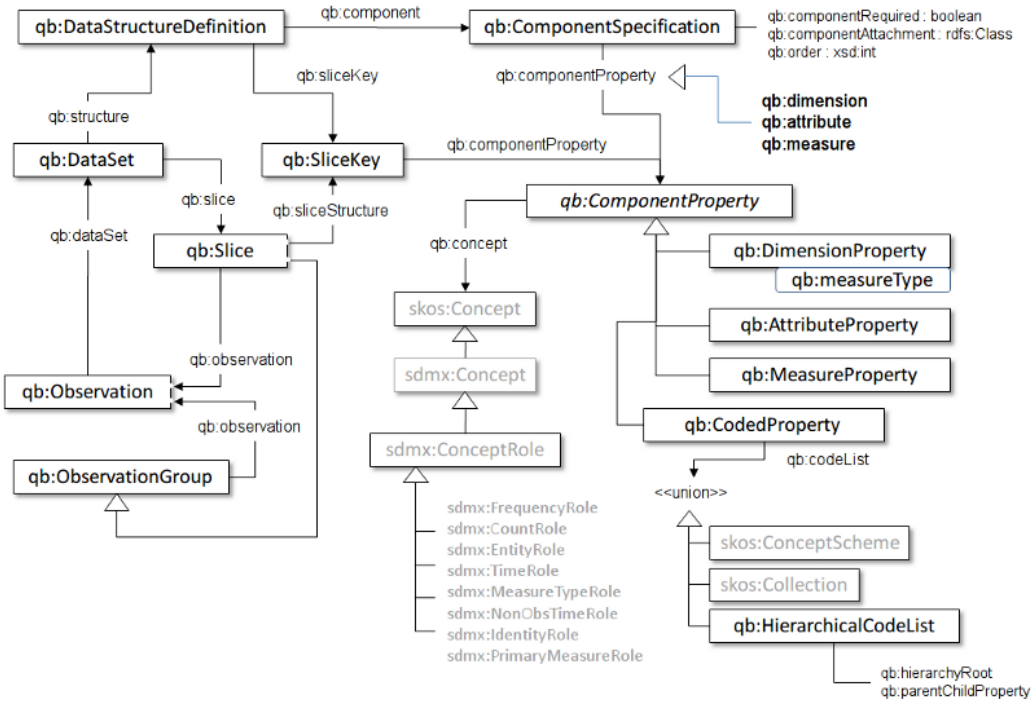
[71] Specifikace dostupná na http://www.w3.org/TR/vocab-org/

[72] Specifikace dostupná na http://sw.joanneum.at/scovo/schema.html

[73] Specifikace dostupná na http://www.w3.org/2004/02/skos/

[74] Specifikace dostupná na http://www.w3.org/TR/void/

jednotlivých „pozorování“,[75] množina těchto pozorování se ukládá do datasetů. ***Obrázek 8*** popisuje architekturu Data Cube Vocabulary (tedy definované třídy a vlastnosti). Některé z těchto prvků jsou popsány v následujících podkapitolách.



Obrázek 8:

Architektura Data Cube Vocabulary [12]

Data Structure Definition

Jak je uvedeno výše, datová struktura se skládá z komponent, které se definují jako RDF vlastnosti. Vlastnosti by měly být instancí abstraktní třídy *qb:ComponentProperty*. Tato třída má podtřídy pro jednotlivé typy komponent – *qb:DimensionProperty*, *qb:MeasureProperty* a *qb:AttributeProperty*. U jednotlivých vlastností je opět možné definovat anotace (název, komentáře) a určit obor hodnot pomocí *rdfs:range*. Z definovaných komponent lze poté sestavit datovou strukturu. Pro definici datové struktury datasetu slouží třída *qb:DataStructureDefinition*, jejíž instancí bude daná datová struktura. Definice struktury dat je oddělená od dat samotných, proto ji lze využívat pro více datových množin. V definici datové struktury se poté uvedou všechny komponenty. K dimenzím lze navíc doplnit jejich pořadí a k atributům, zda je povinné je uvádět či ne. Dále do ní lze přidat údaj o existenci vytvořených pohledů do struktury pomocí vlastnosti *qb:sliceKey* [12].

[75] „observation“

Pojmy z datové struktury, které se vyskytují opakovaně, lze definovat pomocí slovníku SKOS. K tomu slouží vlastnost *qb:concept*, propojující komponentu (*qb:ComponentProperty*) a pojem ze slovníku SKOS (*skos:Concept*). Pokud se nějaký pojem často opakuje, ale není ve SKOS popsán, je možné ho definovat jako instanci třídy *skos:Concept*. Specifickou množinou konceptů definovaných ve SKOS jsou takzvané číselníky[76] – seznamy hodnot pro určitou doménu (například měna, jazyk, apod.). K tomu slouží vlastnost *qb:codeList*. Obor hodnot této vlastnosti je pak *skos:Concept* [12].

Dataset

Dataset je množina statistických dat, která odpovídá nějaké struktuře. Dle [12] existují čtyři druhy dat, jež se mohou v datasetu vyskytnout:

1. **Jednotlivá pozorování** – klíčová sledovaná data;
2. **Organizační struktura** – data sloužící pro vyhledávání jednotlivých pozorování (například hodnoty dimenzí);
3. **Strukturální metadata** – metadata sloužící pro popis a interpretaci dat – atributy;
4. **Referenční metadata** – metadata popisující celý dataset (informace o kategorizaci, autorovi, apod.).

Samotný dataset lze definovat jako instanci třídy *qb:DataSet*. K tomuto zdroji pak lze připojit informace o dané množině dat, například název (pomocí *rdf:label*), popis (*rdf:comment*), jméno autora (*dc:creator*),[77] a především datovou strukturu (pomocí vlastnosti *qb:structure*). Jednotlivá pozorování se pak definují jako instance třídy *qb:Observation*. Ke každému pozorování se poté připojí hodnoty definovaných dimenzí, atributů a měření (pomocí vlastností, které byly definovány výše) [12].

Slices

V některých případech je vhodné seskupit data v datasetu do podmnožin. K tomu slouží takzvané slices[78], což jsou pohledy do datové struktury, které seskupí data do podmnožin dle hodnot vybrané dimenze. Pro popis, podle jaké dimenze se bude pohled vytvářet, slouží *qb:SliceKey* (pohled bude definován jako její instance). Zbylé dimenze se uvedou jako objekty vlastnosti *qb:componentProperty*. Jednotlivé pohledy se poté popíší jako instance třídy *qb:Slice*, napojí se na strukturu pohledu pomocí vlastnosti *qb:sliceStructure* a připojí se k nim jednotlivá pozorování pomocí *qb:observation* [12].

---

[76] „code lists“

[77] Prefix „dc“ představuje slovník Dublin Core, jehož specifikace je dostupná na http://dublincore.org/

[78] do češtiny lze přeložit jako „řezy“

Informace o datasetech

Pro publikování statistických dat pomocí Data Cube Vocabulary je důležité, aby množiny dat byly popsány dodatečnými informacemi, tedy metadaty. Metadata poté slouží k objevování, prezentování a zpracování těchto datasetů. K připojení dodatečných informací lze použít slovník Dublin Core. V ***Tabulka 2*** jsou popsány pojmy z tohoto slovníku, které by se podle [12] měly objevit v popisu datasetu.

Tabulka 2: *Pojmy definované slovníkem Dublin Core, které je vhodné použít pro popis datasetu [12]*

Název	Popis
dct:title[79]	Název daného zdroje (stejné jako rdfs:label, které je vhodné také uvést kvůli zpětné kompatibilitě)
dct:description	Popis zdroje (možno použít i rdfs:comment pro zpětnou kompatibilitu)
dct:issued	Datum vydání
dct:modified	Datum poslední změny
dct:subject	Předmět datasetu
dct:publisher	Autor
dct:license	Právní dokument popisující, co lze se zdrojem dělat

3.1.2 Doporučený postup návrhu datové struktury pomocí Data Cube Vocabulary

Pokud existuje požadavek na publikaci statistických dat jako Linked Data, je ideální použít slovník Data Cube Vocabulary. Při samotném návrhu datové struktury [30] doporučuje postupovat dle následujících kroků (rozepsaných v následujícím textu):

1. Definice prefixů
2. Vytvoření datové struktury
3. Definice datasetu a jednotlivých pozorování
4. Vytvoření pohledů do datové struktury
5. Výběr vhodného URI

[79] Prefix dct představuje jmenný prostor dostupný na http://purl.org/dc/terms/

Definice prefixů

V prvé řadě je vhodné definovat zdroje, které budou při návrhu datové struktury použity. Často používanými slovníky pro návrh datových struktur jsou RDFS (prefix rdfs) a OWL (prefix owl) pro ontologické konstrukty, pro definici datových typů se často využívá XML Schema (xsd). Import externích slovníků lze zapsat pomocí RDF trojice, která má následující strukturu:

```
prefix „název prefixu“: <„URI slovníku“>.
```

Pro import Data Cube Vocabulary se obvykle používá prefix „qb“:

```
prefix qb: < purl.org/linked-data/cube#> .
```

Pro návrh statistických struktur je důležité vědět o existenci standardu SDMX, který definuje velké množství entit a seznamů hodnot (číselníků). Ty lze využít při návrhu struktur, aniž by bylo potřeba tyto entity znovu popisovat. Například při definici dimenze určující pohlaví osoby lze použít pojem *sdmx-dimension:sex*.[80]

Vytvoření datové struktury

Po zavedení prefixu lze přistoupit k popisu komponent datové struktury. Jednotlivé údaje, které jsou sledovány, se popíší jako měření (budou instancí vlastnosti *qb:MeasureProperty*). Problém může nastat při rozhodování, zda údaj uvést jako dimenzi či jako atribut. Obecně platí, že údaj, který slouží k popisu některého jiného údaje či celého datasetu, je vhodné definovat jako atribut (tedy vlastnost, která bude sloužit jako atribut, bude instancí *qb:AttributeProperty*). Naopak data, která jsou důležitá a se kterými se více pracuje, je dobré definovat jako dimenze (instance *qb:DimensionProperty*). Dimenze identifikují, k čemu se vztahují jednotlivá měření, přičemž lze k nim připojit další užitečné informace (metadata). Na dimenze lze též odkazovat. Jako příklad dimenze lze uvést umístění, čas nebo pohlaví. Nakonec je třeba popsat datovou strukturu, a to pomocí *qb:DataStructureDefinition*. V ní definujeme pořadí jednotlivých dimenzí nebo například prvky, k nimž náleží definované atributy [30].

Definice datasetu a jednotlivých pozorování

Pokud je definována datová struktura, lze vytvořit dataset a jednotlivá pozorování. Dataset slouží pro seskupení souvisejících pozorování a definuje se jako instance třídy *qb:Dataset*. K tomuto datasetu je poté třeba připojit datovou strukturu (k tomu slouží vlastnost

[80] Prefix odkazuje na jmenný prostor http://purl.org/linked-data/sdmx/2009/dimension#

qb:structure). Jednotlivá pozorování jsou instance třídy *qb:Observation*. K nim se připojí konkrétní údaje dimenzí, atributů a měření a také název datasetu, do něhož pozorování spadají. Může se stát, že při vytvoření jednotlivých pozorování autor zpozoruje, že datová struktura není efektivně navržena, například že je u všech pozorování stejná hodnota atributu. To, že je struktura dobře navržena, se potvrdí až po vytvoření dat, občas je tedy třeba se k ní vrátit a změnit ji (například opakující se atribut připojit k celému datasetu místo k pozorování) [30].

Vytvoření řezů datové struktury

Data Cube Vocabulary umožňuje vytvářet skupiny podmnožin jednotlivých pozorování, a to za pomoci takzvaných řezů (slices). Pozorování lze seskupit podle hodnoty vybraných dimenzí (například seskupení podle pohlaví osob). Tato možnost je efektivní především u rozsáhlých datasetů. Jednotlivé pohledy jsou reprezentovány jako instance třídy *qb:Slice*. Nejprve je však potřeba definovat strukturu pohledu, a to propojením datové struktury a struktury pohledu (instance třídy *qb:SliceKey*) pomocí vlastnosti *qb:sliceKey*. Jednotlivá pozorování a struktura se poté přiřadí řezu pomocí vlastnosti *qb:observation* a *qb:sliceStructure* [30].

Výběr vhodného URI

Výběr jmenného prostoru je velmi důležitý, přes něj totiž budou mít ostatní uživatelé přístup k definované struktuře či datům. Data i struktura mohou být uloženy v jednom souboru (a tedy na jediné adrese) nebo mohou být distribuovány do samostatných souborů. URI by pak měl vystihovat povahu obsahu, který pod tímto jmenným prostorem lze najít. Například by mělo být patrné, zda lze pod daným URI najít data a o čem tato data vypovídají [30].

3.1.3 Implementace Data Cube Vocabulary

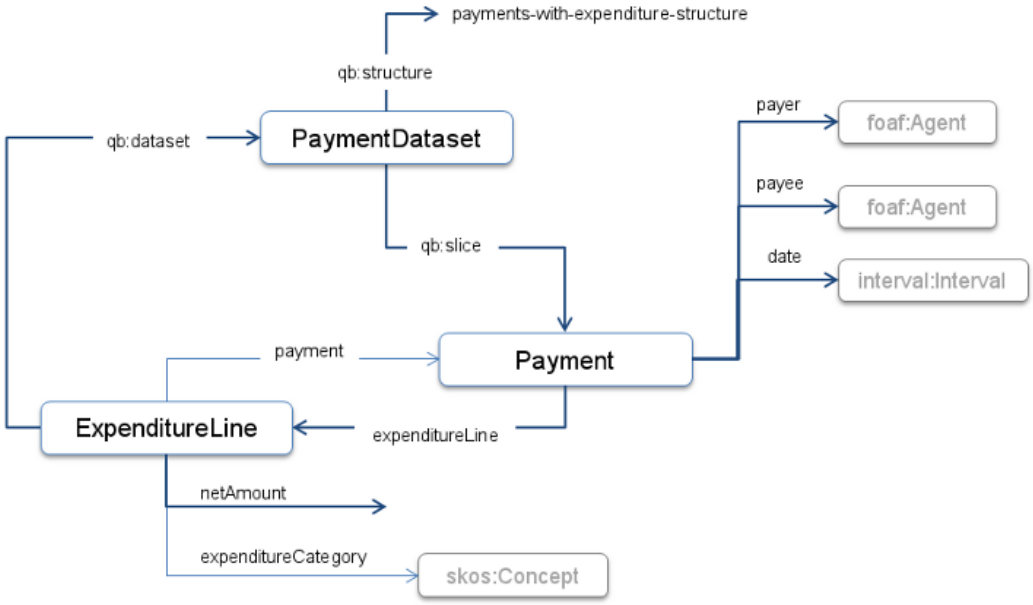
Slovník Data Cube Vocabulary spolu s pojmy definovanými v Content-oriented Guidelines je doporučován pro publikaci statistických dat na Sémantickém webu, je proto poměrně rozšířen. Seznam implementací Data Cube Vocabulary je zveřejňován konsorciem W3C a v současné době (k dubnu 2015) se v něm nachází na 30 projektů.[81] Lze mezi nimi najít několik datasetů obsahující statistické údaje z různých oblastí ve Velké Británii. Za zmínku stojí projekt *„Local Government Payments“*, v rámci nějž jsou publikována data o platbách dodavatelům. Tento projekt se opírá o ontologii nazvanou „*Payments*

[81] Seznam implementací Data Cube Vobaculary je dostupný na http://www.w3.org/2011/gld/wiki/Data_Cube_Implementations

Ontology“,[82] která je rozšířením právě Data Cube Vocabulary. Popisu této ontologie se věnuje následující podkapitola.

Payments Ontology

Ontologie Payments Ontology byla vyvinuta za účelem reprezentace dat o výdajích organizací, tedy nejen pro státní instituce. Obsahuje minimální povinnou strukturu. Základem ontologie je třída „*payment:Payment*“,[83] k níž je připojen plátce a příjemce pomocí vlastností *payment:payer a payment:payee* a také datum (*payment:date*). K platbě lze též připojit další údaje, například organizační jednotku nebo odkazy na popis procesů, které s platbou souvisejí. Architekturu této ontologie popisuje ***Obrázek 9***.



Obrázek 9:
Architektura Payments Ontology [35]

3.1.4 Programy pro práci se statistickými daty

CubeViz

Nástroj CubeViz[84] slouží jako prohlížeč statistických dat využívající slovník Data Cube Vocabulary. Nástroj umožňuje načtená statistická data filtrovat a graficky je zobrazovat,

[82] Popis ontologie lze nalézt na http://data.gov.uk/resources/payments

[83] prefix „payment“ nahrazuje URI http://reference.data.gov.uk/def/payment#

[84] http://aksw.org/Projects/CubeViz.html

například jako lineární nebo sloupcové grafy. Nástroj CubeViz vyvíjí členové komunity AKSW[85] *(Agile Knowledge Engineering and Semantic Web)*. Tato komunita se zabývá výzkumem a vývojem metod a aplikací pro znalostní inženýrství na Sémantickém webu [1]. CubeViz je rozšířením OntoWiki[86], nástroje pro vizuální reprezentaci znalostních bází. Práce v CubeViz lze rozdělit do dvou komponent. První komponenta slouží pro výběr dat pomocí dotazování do strukturální části RDF grafu (tím je myšlena datová struktura popsaná pomocí Data Cube Vocabulary). Druhá, vizualizační komponenta, umožňuje graficky znázorňovat konkrétní pozorování vybraná pomocí první komponenty. Nástroj používá pro dotazování jazyk SPARQL a lze vytvářet výběry dat podle datasetu, pohledu (slice) nebo podle konkrétního měření či atributu nebo dimenze [1].

Payola

Dalším nástrojem pro práci se statistickými daty využívajícími Data Cube Vocabulary je Payola[87]. Aplikace běží ve webovém prostředí a dokáže graficky zobrazovat propojená data pomocí zabudovaných pluginů, například pro zobrazení v grafu nebo tabulce. Také umožňuje vytvářet analýzy a testovat je vůči SPARQL endpointům bez větší znalosti jazyka SPARQL. Payola je vyvíjena na Matematicko-fyzikální fakultě Univerzity Karlovy. Na aplikaci lze nahlížet jako na jako soukromý či podnikový nástroj pro editaci RDF dat, či jako na webovou stránku pro publikování RDF dat [19].

Linked Data Cube Explorer

Nástrojem, který pracuje online, je Linked Data Cube Explorer (LDCX).[88] Slouží k analýze statistických datasetů. Po načtení konkrétního datasetu (ze zadaného URI) lze vybrat jednotlivá pozorování a po nastavení, jak se která dimenze má zobrazit (tedy buď jako sloupec, řádek nebo pohled), program vygeneruje tabulku s jednotlivými údaji.

3.2 Proprietární slovníky pro popis statistických dat

Proprietární slovníky jsou navrženy tak, že popisují pouze uzavřenou oblast reálného světa. Na rozdíl od obecných slovníků, jako je například Data Cube Vocabulary, je velmi omezená jejich možnost použití pro více datasetů. Navíc nejsou kompatibilní s analytickými nástroji, jako je například CubeViz. Mezi proprietární slovníky lze zařadit

---

[85] Více na http://aksw.org

[86] Specifikace nástroje dostupná na http://aksw.org/Projects/OntoWiki.html

[87] http://datacube.payola.cz/

[88] Nástroj je dostupný na http://km.aifb.kit.edu/projects/ldcx/

například ontologii Public Spending. Slovník Public Spending se používá pro popis dat z oblasti veřejných rozpočtů, konkrétně má za cíl propojovat údaje o veřejných výdajích z celého světa. Nevýhodou této ontologie však je, že není postavena na slovníku Data Cube Vocabulary. Data popsaná pomocí Public Spending je tedy možné analyzovat pouze v konkrétním webovém rozhraní tohoto projektu, nikoliv však v obecně rozšířených programech. Aby bylo možné plně využít plného potenciálu propojených dat, je vhodné pro data postavená na této ontologii navrhnout novou datovou strukturu, která se opírá o obecný slovník Data Cube Vocabulary. Tomuto problému se věnuje následující kapitola 4. V rámci této kapitoly je též blíže popsána struktura ontologie Public Spending.

4 Návrh datové struktury pro data z oblasti veřejných rozpočtů

Tato kapitola je věnována popisu navržené datové struktury pro veřejné výdaje. Struktura vychází z výše zmíněné ontologie *Public Spending*. Na rozdíl od Public Spending je v ní použit slovník Data Cube Vocabulary, který je doporučen pro publikaci statistických dat jako Linked Data, a též je možné využít nástrojů pro vizualizaci těchto dat, které byly popsány v předchozí kapitole. Data, která byla původně publikována podle proprietární ontologie Public Spending, mohou být namapovány i na tuto novou strukturu a lze je tedy analyzovat v nástrojích podporujících Data Cube Vocabulary. Podobnou ontologií je *Payments Ontology*, ta je však zaměřena na jiný druh dat. V následující kapitole je blíže popsána struktura ontologie Public Spending a souvisejících dat. Na tomto návrhu lze demonstrovat, jak lze existující data a ontologie převést na podobu, která se opírá o slovník Data Cube Vocabulary.

4.1 Ontologie Public Spending

Jedním z projektů, zabývajícím se veřejnými rozpočty, je Public Spending.[89] Cílem Public Spending je smysluplné propojení dat o veřejných výdajích z celého světa. Data jsou získávána z otevřených zdrojů (jako je DBpedia) a také přímo od státních útvarů, které je zveřejňují z důvodu větší transparentnosti a povědomí společnosti. Tato data jsou následně analyzována, propojena a použita k vykreslení grafů či tabulek, pro lepší přehlednost a vytěžení informací. Projekt staví na technologiích Sémantického webu, zejména principech Linked Open Data a běžných technikách NLP.[90] V současné době je zapojeno sedm státních útvarů, vyjmenovaných v ***Tabulka 3*** spolu se zajímavými údaji o množství zveřejněných informací [33].

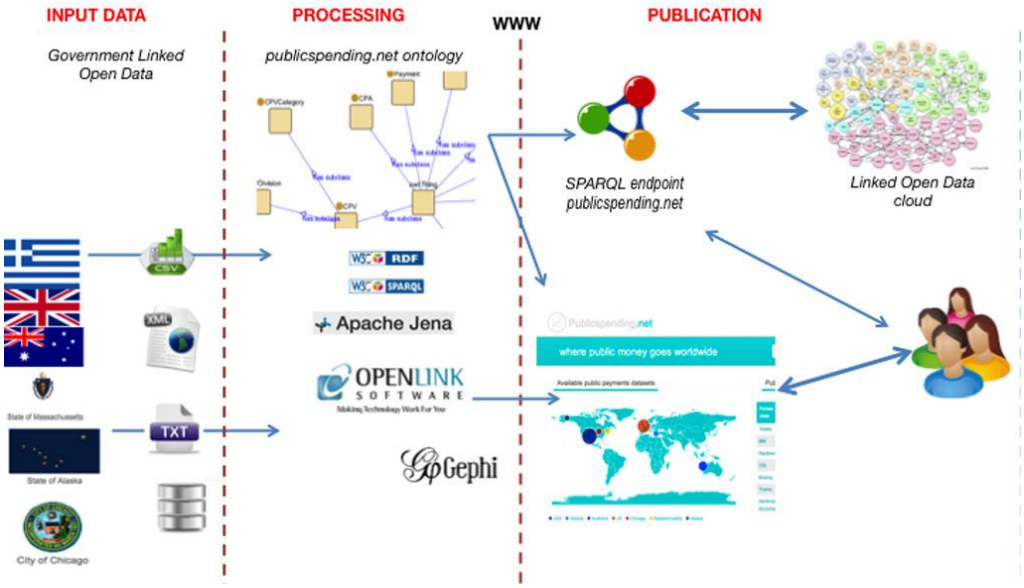
[89] http://publicspending.net/

[90] Natural Language Processing (zpracování přirozeného jazyka) je technologie pro překlad mezi strojovými a lidskými [23].

Tabulka 3: *Státní útvary zapojené do projektu Public Spending spolu s údaji o počtu zaznamenaných plateb a celkové částce k 18. 4. 2015.*

Státní útvar	Počet zaznamenaných plateb	Celková částka
Aljaška	139 821	$ 20 965 616 162
Austrálie	429 435	$ 265 987 872 975
Chicago	84 405	$ 57 399 602 771
Řecko	3 079 905	$ 72 617 857 894
Spojené státy americké	61 359	$ 655 979 790 614
Stát Massachusetts	1 305 267	$ 66 663 356 099
Velká Británie	1 137 641	£ 502 033 843 436

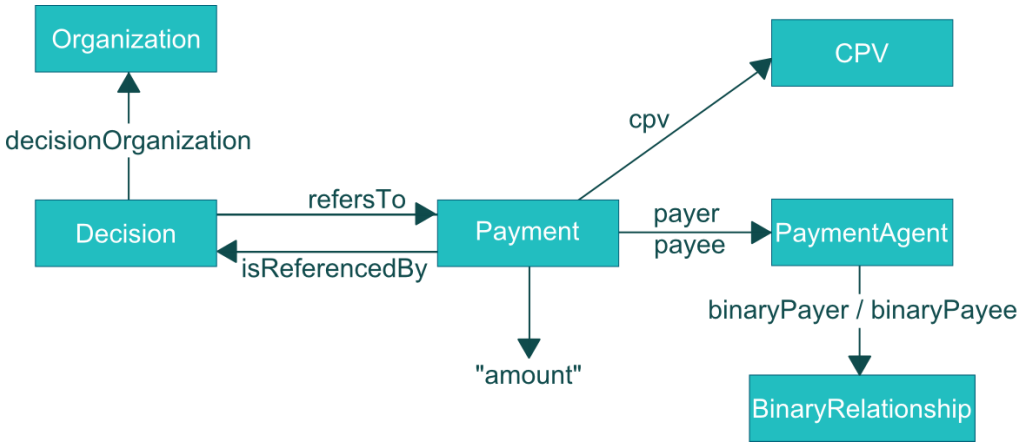
Zpracování získaných dat o veřejných rozpočtech lze rozdělit do tří fází. V první fázi jsou data získávána z různých zdrojů a v různých formátech (CSV, XML, a další textové formáty). Ve fázi zpracování jsou data namapována na ontologii Public Spending, blíže popsanou v následující podkapitole. Poslední fází je zveřejnění dat. Různé přehledy a agregovaná data jsou dostupná na webové stránce projektu, ale k původním informacím lze přistupovat pomocí SPARQL endpointu, do něhož lze provádět různé dotazy. Proces zpracování dat je zobrazen na ***Obrázku 10***.



Obrázek 10:
Proces zpracování dat pomocí Public Spending [33]

Poté, co jsou získána data o veřejných rozpočtech, jsou na základě ontologie Public Spending[91] a dalších rozšířených slovníků (například FOAF nebo Dublin Core) připraveny RDF modely. Dále se pod jednotným URI shromáždí data o společnostech působících ve více zemích. Ontologie se skládá z 15 tříd a 36 vlastností, základní třídy a vlastnosti, tvořící jádro ontologie, jsou popsány v následujícím textu [34].

Pro platební rozhodnutí zde existuje třída *psnet:Decision*,[92] která je spojena s jednotlivými platbami, pro něž existuje třída *psnet:Payment*, pomocí vlastnosti *psnet:refersTo* (inverzní vlastností je *psnet:isReferencedBy*). Třída definující všechny účastníky plateb (tj. plátce a příjemce) se nazývá *psnet:PaymentAgent* (a je ekvivalentní s třídou *foaf:Agent*). Agenti jsou propojeni s platbami pomocí vlastností *psnet:payee* a *psnet:payer* (na základě povahy agenta). Další třídy definují hierarchii CPV kódů. CPV (Common Procurement Vocabulary) je systém pro klasifikaci veřejných zakázek, standardizující odkazy, které jsou používány zadavateli a zadávajícími subjekty k popisu veřejných zakázek [38]. Základní třídou pro popis kódů CPV se nazývá *psnet:CPV*. Obsahuje pět podtříd, první z nich je *psnet:CPVDivision* představující klasifikaci CPV na nejvyšší úrovni (s celkem 45 instancemi). Tuto třídu dále rozšiřuje *psnet:CPVGroup* (obsahující 272 instancí třídy), pro detailnější klasifikaci existují další podtřídy *psnet:CPVClass* s 1002 instancemi, *psnet:CPVCategory* (2 379 instancí) a nakonec *psnet:CPVGround* obsahující 5 756 instancí třídy. CPV kód je spolu s částkou (skrze vlastnost *psnet:paymentAmount*) připojen k *psnet:Payment* pomocí vlastnosti *psnet:cpv*. Dále stojí za zmínku třída *psnet:BinaryRelationship*, reprezentující binární vztahy mezi plátci a příjemci [34]. Pro přehlednější popis vztahů tříd a vlastností byl vytvořen ***Obrázek 11***.



Obrázek 11
Schéma tříd definovaných ontologií Public Spending [32]

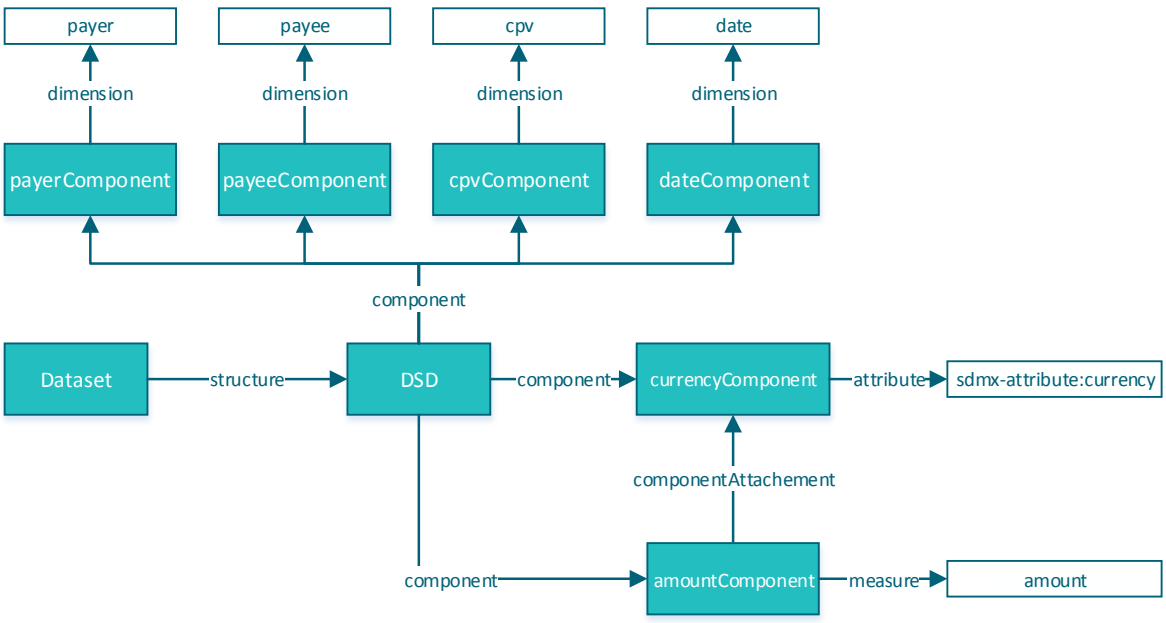
[91] Kompletní dokumentace ontologie je dostupná na https://docs.google.com/document/d/16fxFgtjRZC5AU00RiR0jdzbRFU73cBcOGl8ZZECwl6U/edit?pli=1#heading=h.6dxtxjl2and6

[92] Prefix *psnet* odkazuje na URI http://publicspending.net/ontology#

4.2 Návrh datové struktury – Public Spending Data Cube

Ontologie Public Spending Data Cube byla vyvinuta za účelem publikování dat o veřejných výdajích. Možnost publikovat tato data jako Linked Data otevírá další výhody práce s těmito daty, na rozdíl od zveřejnění těchto dat jinými způsoby (například jako spreadsheety). Tato ontologie umožňuje reprezentovat data o platbách v oblasti veřejných rozpočtů, přičemž jednotlivým platbám jsou přiřazeny jedinečné identifikátory URI, takže jsou odkazovatelné a lze o nich na tomto URI zveřejnit dodatečné údaje. Další výhodou je možnost propojení se souvisejícími datasety. Ontologie Public Spending Data Cube vychází z Public Spending, s tím rozdílem, že využívá obecného slovníku Data Cube Vocabulary. Z toho vyplývá, že data, která mají strukturu definovanou touto ontologií, lze namapovat i na navrženou strukturu a následně je importovat do výše zmíněných analytických nástrojů, popsaných v kapitole 3.1.4. Struktura této ontologie je popsána na ***Obrázku 12***. Celé znění definice struktury lze nalézt v ***Příloze A***.

Ontologie je určena pro uživatele, kteří se již setkali s publikováním informací jako Linked data, a pro lidi, kteří chtějí publikovat údaje v oblasti veřejných rozpočtů tímto způsobem, a to buď jako součást výzkumů nebo při převodu existujících dat na propojená data (způsob mapování existujících dat je popsán v následujícím textu). Tato ontologie je poměrně otevřená, lze ji proto použít v návrzích nových datových struktur, které by ji dále rozšiřovaly. Pro lepší čitelnost byla použita notace Turtle.



Obrázek 12:
Struktura ontologie Public Spending Data Cube

Ontologie zahrnuje definici čtyř dimenzí, které popisují jednotlivé platby – plátce (*psdc:payer*), příjemce (*psdc:payee*), datum (*psdc:date*) a CPV kód specifikující oblast,

která se týká dané platby (*psdc:cpv*). K definici jednotlivých CPV kódů byl zvolen code list „*CPV 2008 Linked Open Data*“.[93] Sledovanou hodnotu poté reprezentuje *psdc:amount*, tedy konkrétní částka realizované platby. S ní souvisí i atribut představující měnu (*psdc:currency*), definovaný pomocí SDMX atributu.

4.2.1 Volba jmenného prostoru a prefixu

Jako jmenný prostor ontologie byl zvolen identifikátor URI:

```
 < https://raw.githubusercontent.com/lindahorakova/public-spending-data-
cube/master/psdc.ttl#>
```

Na tomto URI je ontologie dostupná pro prohlížení. Formát URI je zvolen tímto způsobem, protože odkazuje na soubor, ve kterém je uložena jak celá ontologie, tak znalostní báze. Pro popis jednotlivých prvků ontologie v kódu byl vybrán prefix „psdc“, jakožto zkratka názvu *„Public Spending Data Cube*“. Při implementaci této ontologie do návrhu jiných datových je však možné používat jiné prefixy (to je dáno povahou RDF, tedy prefix je volitelný), tento název je však doporučené používat.

4.2.2 Datová struktura

Datová struktura je definována jako instance třídy *qb:DataStructureDefinition* a je nazvána *psdc:DSD*. Jednotlivé komponenty datové struktury jsou na ni napojeny skrze vlastnost *qb:component*. Tyto komponenty jsou instancemi třídy *qb:ComponentSpecification*. Jednotlivé dimenze, měření a atributy jsou v Data Cube Vocabulary definovány jako vlastnosti a jsou instancemi podtříd třídy *qb:ComponentProperty*, v závislosti na typu komponenty. ***Tabulka 4*** blíže popisuje jednotlivé vlastnosti použití při návrhu datové struktury. Datová struktura je v ontologii definována takto:[94]

Výpis 5: *Definice datové struktury*

```
psdc:DSD   a      qb:DataStructureDefinition;
                  qb:component psdc:payerComponent, psdc:payeeComponent,
                  psdc:dateComponent, psdc:cpvComponent, psdc:AmountComponent,
                  psdc:currencyComponent;
                  .
```

[93] více na http://cpv.data.ac.uk/

[94] Pro přehlednost ukázkových kódů byly z kódu odebrány anotace, tedy popisy a názvy objektů.

Tabulka 4: *Přehled vlastností používaných pro návrh datové struktury [12]*

Název vlastnosti	Typ vlastnosti	Definiční obor (qb:domain)	Obor hodnot (qb:range)
psdc:payer	qb:DimensionProperty	qb:Observation	psdc:PaymentAgent
psdc:payee	qb:DimensionProperty	qb:Observation	psdc:PaymentAgent
psdc:date	qb:DimensionProperty	qb:Observation	xsd:date
psdc:cpv	qb:DimensionProperty	qb:Observation	skos:Concept
psdc:amount	qb:MeasureProperty	qb:Observation	xsd:decimal

Prvek *psdc:payerComponent* je název komponenty představující plátce dané platby, obvykle je to státní instituce vystupující v platebním vztahu jako poskytovatel peněžních prostředků. Pomocí vlastnosti *qb:dimension* je ke komponentě připojena vlastnost *psdc:payer*, která slouží k určení konkrétního plátce. Komponenta a vlastnost *psdc:payer* jsou v ontologii definovány takto:

Výpis 6: *Definice plátce*

```
psdc:payerComponent a                  qb:ComponentSpecification;
                    qb:dimension       psdc:payer;
                    qb:order           1.

psdc:payer          a                  rdf:Property, qb:DimensionProperty;
                    rdfs:subPropertyOf psnet:payer;
                    rdfs:domain        qb:Observation;
                    rdfs:range         psdc:PaymentAgent;
                    .
```

Komponenta *psdc:payeeComponent* představuje příjemce platby, tedy společnost, která je financována například státní institucí. Jako v předchozím případě se jedná o dimenzi, je k ní tedy napojena skrze *qb:dimension* vlastnost *psdc:payee*. Tato vlastnost se pak použije k připojení URI příjemce k platbě.

Výpis 7: *Definice příjemce*

```
psdc:payeeComponent a                  qb:ComponentSpecification;
                    qb:dimension       psdc:payee;
                    qb:order           2 .
psdc:payee          a                  rdf:Property, qb:DimensionProperty;
```

```
                    rdfs:subPropertyOf psnet:payee;
                    rdfs:domain        qb:Observation;
                    rdfs:range         psdc:PaymentAgent;
                    .
```

Obor hodnot *psdc:payer a psdc:payee* jsou instance třídy *psdc:PaymentAgent*, což je třída speciálně definovaná pro účastníky jednotlivých plateb. Obor hodnot je pro tyto vlastnosti stejný, jelikož jeden účastník může v různých platbách vystupovat jako plátce i jako příjemce. Následuje konkrétní definice třídy *psdc:PaymentAgent*.

Výpis 8: *Definice agenta*

```
psdc:PaymentAgent   a rdfs:Class;
                    rdfs:subClassOf  psnet:PaymentAgent;
                    rdfs:subClassOf  foaf:Agent;
                    .
```

Další komponentou je datum (*psdc:dateComponent*). Jako v předchozích dvou případech se jedná o dimenzi, konkrétně popisující, kdy proběhla daná platba. Vlastnost *psdc:date* připojuje k dimenzi údaj o datu ve formě literálu, který je typu *xsd:date*.

Výpis 9: *Definice času*

```
psdc:dateComponent  a                qb:ComponentSpecification;
                    qb:dimension     psdc:date;
                    qb:order         3 .

psdc:date           a owl:DatatypeProperty, qb:DimensionProperty;
                    rdfs:subPropertyOf  sdmx-dimension:refTime;
                    rdfs:domain         qb:Observation;
                    rdfs:range          xsd:date;
                    .
```

Poslední dimenze představuje CPV kód, specifikující, jaké oblasti se platba týká. Komponenta popisující CPV kód se nazývá *psdc:cpvComponent*, vlastnost pro připojení konkrétního kódu se jmenuje *psdc:cpv*. Oborem hodnot této vlastnosti jsou kódy CPV

z code listu „*CPV 2008 Linked Open Data*“, [95]ve kterém lze najít jednotlivé kódy oblastí. V případě, že je oborem hodnot code list, je dobré uvést jeho URI skrze vlastnost qb:codeList. Jako objekt se tedy dle specifikace Data Cube Vocabulary u vlastnosti *rdfs:range* uvede *skos:Concept* [12].

Výpis 10: *Definice CPV*

```
psdc:cpvComponent  a                  qb:ComponentSpecification;
                   qb:dimension       psdc:cpv;
                   qb:order           4 .

psdc:cpv           a                  owl:ObjectProperty, qb:DimensionProperty;
                   rdfs:subPropertyOf psnet:cpv;
                   rdfs:domain        qb:Observation;
                   qb:codeList        <http://purl.org/cpv/2008/scheme-CPV2008>;
                   rdfs:range         skos:Concept;
                   .
```

Samotné měření, tedy konkrétní částka platby, je specifikováno pomocí komponenty *psdc:amountComponent*. K *psdc:amountComponent* je připojena skrze *qb:measure* vlastnost *psdc:amount*. Ke komponentě je též připojen atribut představující měnu (toto připojení je nastaveno defaultně, v případě, že by bylo žádoucí atribut připojit k celému datasetu globálně, bylo by to třeba uvést pomocí vlastnost *qb:componentAttachement*). Atribut je definován pomocí již vytvořeného atributu pro měnu, převzatého ze standardu SDMX (sdmx-attribute:currency).[96] Tento atribut by mohl být připojen k samotnému datasetu, přičemž by hodnota tohoto atributu platila pro všechna zahrnutá pozorování. Z důvodu variability (kdy jednotlivé platby v datasetu mohou mít různou měnu) byl však připojen k jednotlivým platbám.

[95] Pro tento účel by bylo možné použít code list http://linked.opendata.cz/resource/concept-scheme/cpv-2008, v této práci byl však zvolen CPV 2008 Linked Open Data pro svou lepší uživatelskou přívětivost ve webovém rozhraní.

[96] Dokumentace ke code listu použitému ve specifikaci sdmx-attribute:currency je dostupná na http://sdmx.org/wp-content/uploads/2009/01/01_sdmx_cog_annex_1_cdc_2009.pdf

Výpis 11: *Definice měření*

```
psdc:AmountComponent    a                   qb:ComponentSpecification;
                        qb:measure          psdc:amount;

psdc:currencyComponent  a                   qb:ComponentSpecification;
                                            qb:attribute sdmx-attribute:currency
.

psdc:amount             a                   rdf:Property, qb:MeasureProperty;
                        rdfs:subPropertyOf  sdmx-measure:obsValue;
                        rdfs:range          xsd:decimal;
.
```

4.2.3 Ukázková data

Pro ilustraci využití navržené ontologie byla vybrána data s datasetu „*Food and Environment Research Agency*“,[97] který využívá ontologii Public Spending. Na těchto datech bude prezentováno, jak lze data strukturovaná pomocí Public Spending ontology namapovat na ontologii Public Spending Data Cube. Vybrané údaje jsou přehledně zapsány v ***Tabulka 5***. Původní data jsou však uvedena ve formátu RDF/XML, následuje ukázka jednoho záznamu o platbě v původním formátu (RDF/XML) a pro srovnání ve formátu Turtle.

Výpis 12: *Ukázková data v RDF/XML*

```
<rdf:Description
rdf:about="http://publicspending.medialab.ntua.gr/resource/uk/payments/23370-1">
    <psnet:cpv
rdf:resource="http://publicspending.medialab.ntua.gr/resource/cpvCodes/22452000-
3"/>
    <psnet:paymentAmount>166052.36</psnet:paymentAmount>
    <psnet:transaction>
    <rdf:Description
rdf:about="http://publicspending.medialab.ntua.gr/resource/uk/transactions/23370
">
    <psnet:submissionTimeStamp>2004-12-
01T16:08:20.000+02:00</psnet:submissionTimeStamp>
    </rdf:Description>
    </psnet:transaction>
    <psnet:payee>
```

[97] celý původní dataset lze stáhnout na http://datahub.io/en/dataset/publicspending-net/resource/a1f3f38d-15c4-470d-a447-bfc78389151d, konkrétně se jedná o soubor s názvem fera-over-25k-1004.csv10.rdf

```
    <rdf:Description
rdf:about="http://publicspending.medialab.ntua.gr/resource/uk/suppliers/National
_Westminster_Bank_Plc">
    <rdfs:label>National Westminster Bank Plc</rdfs:label>
    </rdf:Description>
    </psnet:payee>
    <psnet:payer>
    <rdf:Description
rdf:about="http://publicspending.medialab.ntua.gr/resource/uk/entities/Fera">
    <rdfs:label>Fera</rdfs:label>
    </rdf:Description>
    </psnet:payer>
    </rdf:Description>
```

Výpis 13: *Ukázková data v Turtle*

```
< http://publicspending.medialab.ntua.gr/resource/uk/payments/23370-1>
    psnet:cpv <
http://publicspending.medialab.ntua.gr/resource/cpvCodes/22452000-3>;
    psnet:paymentAmount 166052.36;
    psnet:transaction <
http://publicspending.medialab.ntua.gr/resource/uk/transactions/23370>;
    psnet:payee <
http://publicspending.medialab.ntua.gr/resource/uk/suppliers/National_Westminste
r_Bank_Plc>;
    psnet:payer <
http://publicspending.medialab.ntua.gr/resource/uk/entities/Fera >
.

< http://publicspending.medialab.ntua.gr/resource/uk/transactions/23370>
    psnet:submissionTimeStamp 2004-12-01T16:08:20.000+02:00 .

<http://publicspending.medialab.ntua.gr/resource/uk/suppliers/National_Westminst
er_Bank_Plc>
    rdfs:label "National Westminster Bank Plc" .

< http://publicspending.medialab.ntua.gr/resource/uk/entities/Fera>
    rdfs:label " Fera " .
```

Tabulka 5: *Konkrétní data definovaná pomocí Public Spending*

payer	payee	cpv	paymentAmount	submissionTimeStamp
Fera	National Westminster Bank Plc	22452000-3	166052.36	2004-12-01T16:08:20.000+02:00
Fera	Forestry Commission	79342100-4	287549.39	2005-11-01T16:08:20.000+02:00
Fera	Forestry Commission	79342100-4	47459.95	2005-07-01T16:08:20.000+02:00
Fera	Innoventors Ltd	79342100-4	35250.73	2006-05-01T16:08:20.000+02:00
Fera	'veterinary Laboratories Agency	79342100-4	41178.06	2005-02-01T16:08:20.000+02:00
Fera	Innoventors Ltd	79342100-4	32556.72	2006-05-01T16:08:20.000+02:00
Fera	Ni-co	79342100-4	46053.01	2004-07-01T16:08:20.000+02:00
Fera	Ip Pragmatics Limited	79342100-4	52081.01	2004-09-01T16:08:20.000+02:00
Fera	University Of Nottingham	79342100-4	25000.0	2005-02-01T16:08:20.000+02:00

Nejprve je třeba zjistit, jak dané entity datasetu korespondují s definovanými entitami vybrané ontologie. V ukázkovém případě je to zřejmé, prvek s názvem *psnet:payer* lze definovat jako *psdc:payer*, *psnet:payee* jako *psdc:payee*, atd. V případě CPV je možné z URI daného kódu zjistit konkrétní hodnotu a vyhledat ji v použitém code listu ontologie. Částka (*psnet:paymentAmount*) je ve stejném formátu jako *psdc:amount*. Měna je stejná pro celý dataset, takže by bylo vhodné připojit atribut měny přímo k definici datasetu, avšak v ontologii je připojen k jednotlivým částkám. Problém může nastat s formátem data, kde ontologie Public Spending využívá časové značky (ve formátu „YYYY-MM-DDTHH:MM:SS.SSS“). Z tohoto údaje však lze extrahovat pouze datum, které je zapsáno stejně jako v případě použití datového typu *xsd:date*.

Dalším krokem je vytvoření datasetu, tedy pojmenované množiny pozorování. Pro ukázkový dataset byl zvolen název *psdc:Dataset*. Je vhodné k datasetům připojovat anotace, tedy metadata popisující množinu dat (viz kapitola 3.1.2). Tyto informace umožní nalezitelnost celé množiny dat, což podpoří jejich znovuvyužití. Dále je nutné datasetu připojit nadefinovanou datovou strukturu, a to skrze vlastnost *qb:structure*. Následuje definice ukázkového datasetu, včetně veškerých informací (*rdfs:label* a *rdfs:comment* jsou uvedeny z důvodu zpětné kompatibility).

Výpis 14: *Definice datasetu*

```
psdc:Dataset a    qb:DataSet;
  rdfs:label      "Platby Food and Environment Research Agency nad 25 000 liber"@cs;
  rdfs:label      "Payments made by Food and Environment Research Agency over 25 000
                  pounds"@en;
  dc:title        "Payments between payers and peyess in public spending"@cs;
  dct:description "Dataset contains data about public spending payments, made by Food
                   and Environment Research Agency."@en;
  rdfs:comment    "Dataset contains data about public spending payments, made by Food
                  and Environment Research Agency."@en;
  dct:issued      "2015-04-18"^^xsd:date;
  dct:modified    "2015-04-18"^^xsd:date;
  dct:subject     "Payments between payers and payees"^^xsd:string;
  dct:publisher   "Linda Horakova"^^xsd:string;
  qb:structure    psdc:DSD;
  .
```

Každé pozorování se definuje jako instance třídy *qb:Observation*. Pomocí výše zmiňovaných vlastností (*psdc:payer, psdc:payee, psdc:date, psdc:cpv, psdc:amount*) lze k pozorování připojit konkrétní hodnoty jednotlivých komponent. Pro pozorování v rámci této ukázky byl zvolen jednotný název „*paymentX*“. Jednotlivé platby zapsané v ***Tabulka 5*** pak lze zapsat jako propojená data s využitím Data Cube Vocabulary a navržené ontologie Public Spending Data Cube. Následuje ukázka vybraných dat (veškerá data lze nalézt v ***Příloze B***).

Výpis 15: *Namapovaná data*

```
psdc:payment1 a qb:Observation;
    qb:dataSet              psdc:Dataset;
    psdc:amount             166052.36;
    sdmx-attribute:currency <http://ecb.270a.info/code/1.0/CL_CURRENCY/GBP>;
```

```
    psdc:payer
    <http://publicspending.medialab.ntua.gr/resource/uk/entities/Fera>;
    psdc:payee
<http://publicspending.medialab.ntua.gr/resource/uk/suppliers/National_Westminst
er_Bank_Plc>;
    psdc:date               "2004-12-01";
    psdc:cpv                <http://purl.org/cpv/2008/code-22452000-3>;
.


psdc:payment2 a qb:Observation;
    qb:dataSet              psdc:Dataset;
    psdc:amount             287549.39;
    sdmx-attribute:currency <http://ecb.270a.info/code/1.0/CL_CURRENCY/GBP>;
    psdc:payer
    <http://publicspending.medialab.ntua.gr/resource/uk/entities/Fera>;
    psdc:payee
    <http://publicspending.medialab.ntua.gr/resource/uk/suppliers/Forestry_Commi
ssion>;
    psdc:date               "2005-11-01";
    psdc:cpv                <http://purl.org/cpv/2008/code-79342100-4>;
    .
```

4.3 Generování dat z existujících zdrojů pomocí SPARQL

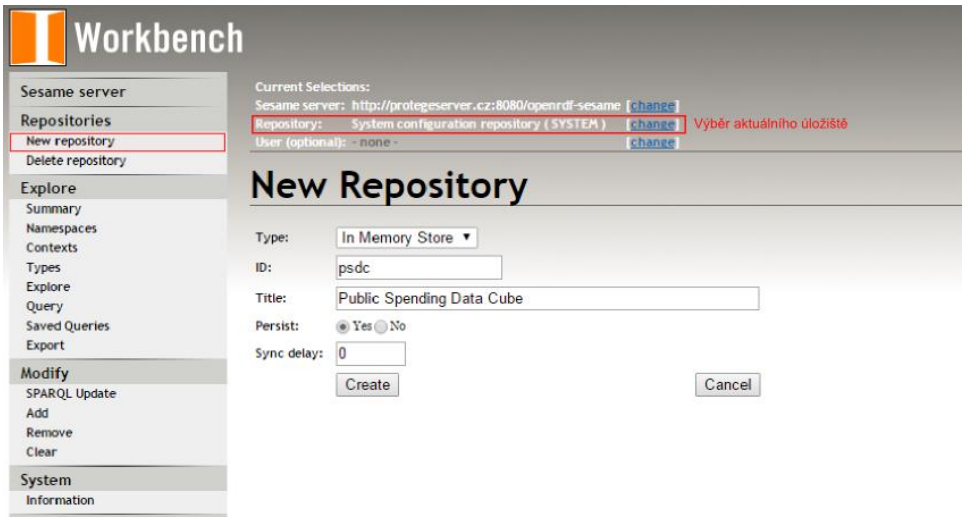
Pro získání dat, která mají strukturu definovanou ontologií Public Spending Data Cube, není třeba data vytvářet ručně. Elegantnějším způsob je vygenerování dat pomocí dotazů SPARQL, konkrétně pomocí příkazů CONSTRUCT[98] nebo INSERT.[99] Rozdíl mezi těmito dvěma příkazy spočívá v tom, že CONSTRUCT vytváří RDF graf podle zadané šablony, přičemž INSERT vkládá data (definovaná šablonou) přímo do daného úložiště. Pro mapování dat na ontologii by bylo vhodnější použít druhou z klauzulí, ta byla ovšem zavedena až s novější verzí SPARQL 1.1, tudíž ji nepodporují všechny triple-story. V následujícím textu lze najít ukázky obou typů dotazu.

[98] http://www.w3.org/TR/rdf-sparql-query/#construct

[99] více na http://www.w3.org/TR/sparql11-update/#deleteInsert

4.3.1 Práce v prostředí Sesame a návrh SPARQL dotazu

Jako prostředí pro vytváření nového RDF grafu byl zvolen framework Sesame[100], grafické prostřední pro dotazování a další funkcionality poskytuje aplikace OpenRDF Workbench.[101] Nejprve bylo nutné nahrát data,[102] která budou transformována. K tomu bylo potřeba vytvořit nové úložiště (v tomto případě s názvem Public Spending Data Cube a ID „psdc“). Do něj byla nahrána ukázková data, která byla též použita pro mapování na ontologii v kapitole 4.2.3. Postup vytváření nového úložiště je zobrazen na ***Obrázku 13.***



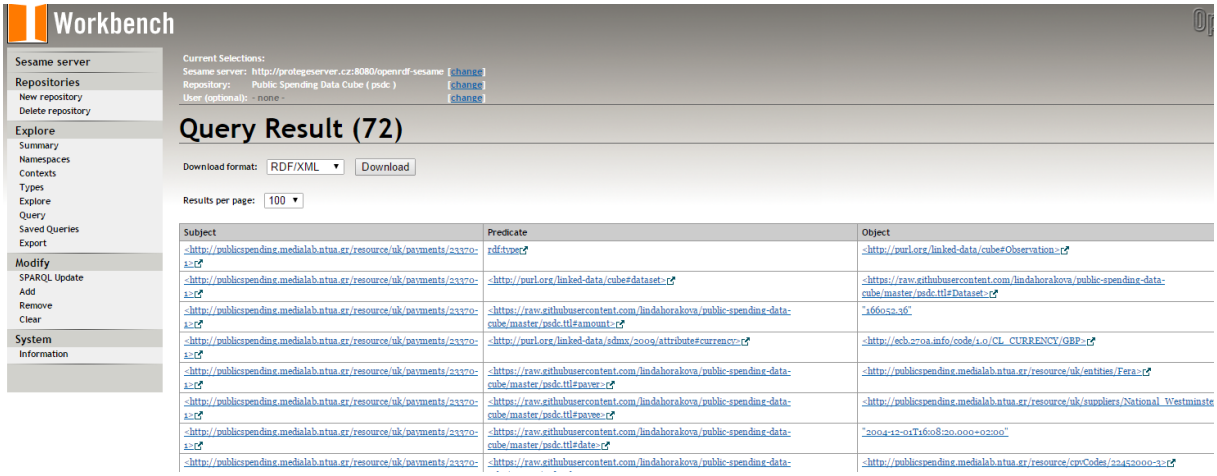
Obrázek 13:
Vytváření úložiště v prostředí Sesame

Od této chvíle je možné vytvářet různé SPARQL dotazy do nahraného RDF grafu. Ty je možné provádět v záložce Query. Je důležité, aby v horní části okna byl zvolen správný zdroj. Ve střední části okna je poté možné zapisovat SPARQL dotazy, ale také nastavit limitní počet výsledků a další možnosti. Výsledek dotazování (viz ***Obrázek 14***) se zobrazí v podobě tabulky se sloupci se subjekty, predikáty a objekty. Tento Výsledný graf je poté možné exportovat do různých formátů, například i do Turtle.

[100] Dokumentace dostupná na http://rdf4j.org/

[101] Dokumentace na http://rdf4j.org/sesame/2.8/docs/articles/workbench.docbook?view

[102] Data jsou dostupná na http://protegeserver.cz:8080/openrdf-workbench/repositories/psnetDCV/query



Obrázek 14:
Výsledek SPARQL dotazu v prostředí Sesame

V tomto konkrétním případě lze použít dotaz CONSTRUCT, který generuje pro každou hodnotu proměnné „*?observation*“ (na místě subjektu) osm trojic, které přesně odpovídají struktuře definované v kapitole 6.2. Na místa objektů v trojicích jsou dosazeny údaje z původního RDF grafu (to lze provést skrze proměnné použité v klauzuli CONSTRUCT a poté v klauzuli WHERE). K většině objektů bylo snadné přistoupit pomocí jediné trojice, v případě údaji o čase však bylo třeba připojit i prvek *psnet:transaction*, protože objekt popisující platbu neodkazuje přímo na časovou hodnotu. Celý SPARQL dotaz vypadá následovně:

Výpis 16: *SPARQL dotaz pro konstrukci RDF grafu*

```
PREFIX dc:              <http://purl.org/dc/terms/>
PREFIX rdfs:            <http://www.w3.org/2000/01/rdf-schema#>
PREFIX rdf:             <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX qb:              <http://purl.org/linked-data/cube#>
PREFIX psnet:           <http://publicspending.medialab.ntua.gr/ontology#>
PREFIXpsdc:
    <https://raw.githubusercontent.com/lindahorakova/public-spending-data-
cube/master/psdc.ttl#>
PREFIX sdmx-attribute:  <http://purl.org/linked-data/sdmx/2009/attribute#>

CONSTRUCT
{
    ?observation    a                           qb:Observation .
    ?observation    qb:dataset                  psdc:Dataset .
    ?observation    psdc:amount                 ?amount .
    ?observation sdmx-attribute:currency
    <http://ecb.270a.info/code/1.0/CL_CURRENCY/GBP> .
    ?observation    psdc:payer                  ?payer .
```

```
    ?observation    psdc:payee                  ?payee .
    ?observation    psdc:date                   ?date .
    ?observation    psdc:cpv                    ?cpv
}
WHERE
{
    ?observation    psnet:paymentAmount         ?amount .
    ?observation    psnet:payee                 ?payee .
    ?observation    psnet:payer                 ?payer .
    ?observation    psnet:cpv                   ?cpv .
    ?observation    psnet:transaction           ?transaction .
    ?transaction    psnet:submissionTimeStamp   ?date
}
```

Při ne příliš složité modifikaci lze dotaz použít i pro jiné datasety, které využívají ontologii Public Spending Ontology. Výstup tohoto dotazu je praktický totožný s manuálně vytvořenými daty v předchozí kapitole, s rozdílem, že je použit jiný code list pro CPV kódy. Aby se ve výsledku zobrazily CPV kódy z code listu definovaném v datové struktuře, bylo by třeba zkonstruovat složitější dotaz, ve kterém by byly použity funkce pro transformaci řetězců. Tento problém může být předmětem vylepšení práce do budoucna. Ačkoliv je v původní struktuře použit pro formát času *xsd:string*, přesto se obsah uloží do formátu data (*xsd:date*).

Na podobném principu funguje i příkaz INSERT, s tím rozdílem, že vygenerovaná data ukládá přímo do zvoleného úložiště. Není tedy třeba exportovat data do souboru a pak je znovu nahrávat na Web. Příkaz INSERT však v mnoha implementacích zatím není podporován, ačkoliv se jedná o rychlejší cestu k získání dat. Pro práci s daty vzniklými v rámci této bakalářské práce bylo však uvolněno prostředí, které příkaz INSERT podporuje. Příkaz lze vkládat v záložce „SPARQL update“. Následující kód představuje celé znění dotazu, přičemž jeho výsledek je shodný s výsledkem dotazu CONSTRUCT uvedeného v předchozím textu.

Výpis 17: *SPARQL dotaz pro vložení nových dat do triplestoru*

```
PREFIX dc:              <http://purl.org/dc/terms/>
PREFIX rdfs:            <http://www.w3.org/2000/01/rdf-schema#>
PREFIX owl:             <http://www.w3.org/2002/07/owl#>
PREFIX rdf:             <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX qb:              <http://purl.org/linked-data/cube#>
PREFIx psnet:           <http://publicspending.medialab.ntua.gr/ontology#>
```

```
PREFIX psdc:
    <https://raw.githubusercontent.com/lindahorakova/public-spending-data-
cube/master/psdc.ttl#>
PREFIX sdmx-attribute:    <http://purl.org/linked-data/sdmx/2009/attribute#>

INSERT
{ GRAPH <http://protegeserver.cz:8080/openrdf-sesame/repositories/psdc>
{
    ?observation         a                        qb:Observation .
    ?observation         qb:dataset               psdc:Dataset .
    ?observation         psdc:amount              ?amount .
    ?observation sdmx-attribute:currency
    <http://ecb.270a.info/code/1.0/CL_CURRENCY/GBP> .
    ?observation         psdc:payer               ?payer .
    ?observation         psdc:payee               ?payee .
    ?observation         psdc:date                ?date .
    ?observation         psdc:cpv                 ?cpv
}
}
WHERE
{
    ?observation         psnet:paymentAmount      ?amount .
    ?observation         psnet:payee              ?payee .
    ?observation         psnet:payer              ?payer .
    ?observation         psnet:cpv                ?cpv .
    ?observation         psnet:transaction        ?transaction .
    ?transaction         psnet:submissionTimeStamp ?date
}
```

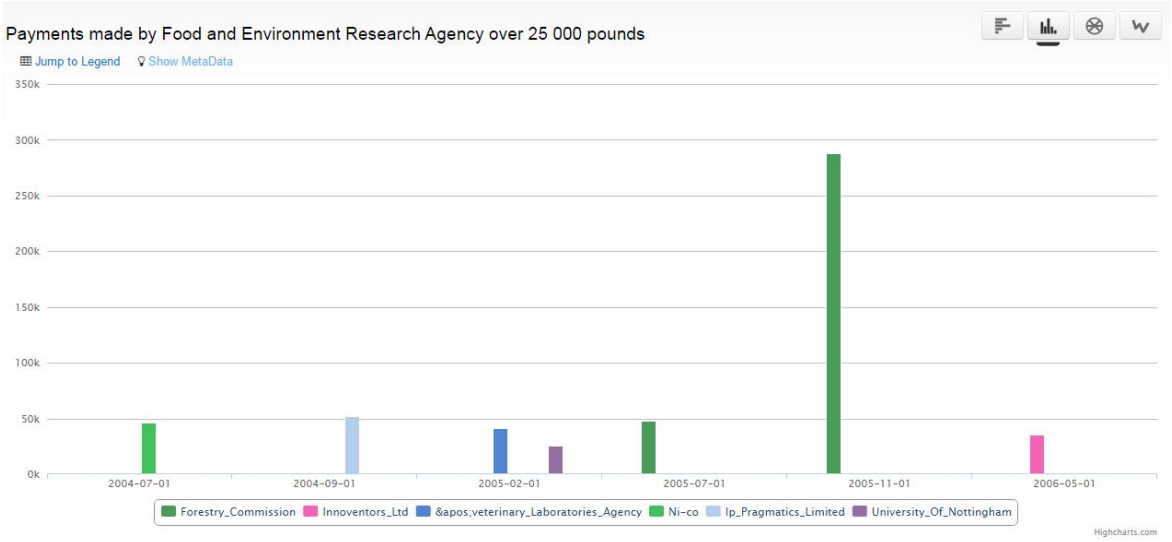
4.4 Zpracování statistických dat v nástroji CubeViz

Data, která vznikla pomocí dotazu uvedeného v předchozí kapitole, je možné naimportovat do nástroje CubeViz[103] (nástroj byl popsán v kapitole 3.1.4). To je možné provést přes prostředí Virtuoso Conductor, protože program CubeViz umí pracovat pouze s tímto prostředím. Zde se nejdříve vytvoří a pojmenuje graf a následně se nahrají data (buď ze souboru nebo z online zdroje). Tato data se poté dají automaticky zpracovávat v programu CubeViz.

Pro zobrazení dat je nejprve nutné vybrat požadovanou znalostní bázi (to lze provést v levém sloupci na hlavní stránce nástroje). V dalším kroku je třeba nastavit, se kterými údaji se bude pracovat (tedy, které hodnoty dimenzí, atributů a měření budou

[103] Nástroj s naimportovanými daty je dostupný na http://protegeserver.cz/cbv/.

využity). Kliknutím na tlačítko „Update chart“ se spustí zpracování dat. Výsledkem je graf zobrazující vybraná data (viz ***Obrázek 15***).



Obrázek 15:
Vizualizace dat v nástroji CubeViz

Na ***Obrázku 15*** lze vidět vizualizaci znalostní báze, která byla vytvořena v předchozí podkapitole. Graf zobrazuje jednotlivé výše plateb daným firmám za určité období. V pravém horním rohu lze přepínat typy grafů, které se mají zobrazit, v tomto případě je nejvhodnější vybrat graf sloupcový. Najetím na konkrétní sloupec se zobrazí dodatečná data, konkrétně CPV kód a přesná částka dané platby. Klikáním na legendu v dolní části lze skrýt jednotlivé příjemce plateb. Pod vygenerovaným grafem jsou zobrazeny další možnosti práce s daty. Daná data je možné exportovat do Turtle či formátu CSV (pravý horní roh). Dále je zde možnost zobrazit dodatečné informace o výstupních datech.

Závěr

Na základě analýzy existujících slovníků Sémantického webu pro popis statistických dat lze dojít k závěru, že je vhodnější využívat obecný slovník Data Cube Vocabulary než proprietární slovníky, které jsou určené pouze pro omezenou doménu, a tím pádem je limitována jejich schopnost vícenásobného využití. Také analýza a další zpracování dat je možné pouze v aplikacích zpravidla určených pro danou ontologii. Na rozdíl od proprietárních slovníků, použití Data Cube Vocabulary umožňuje znalostní báze zpracovávat ve větším množství nástrojů, například v programu CubeViz či Payola. To bylo demonstrováno na ukázkovém příkladu, jehož podstatou bylo převést ontologii Public Spending a související data na datovou strukturu využívající právě Data Cube Vocabulary. Tato struktura by v budoucnu měla být doplněna o řezy (slices), například podle jednotlivých CPV kódů. Znalostní báze Public Spending byla převedena na danou strukturu pomocí SPARQL dotazu. Tato data pak byla naimportována do nástroje CubeViz, aby bylo předvedeno, že použití Data Cube Vocabulary při návrh ontologií pro statistická data usnadňuje využití těchto dat. Dotaz lze v budoucnu vylepšit o funkce, které by automaticky převáděly URI code listů na ty, které jsou definovány v datové struktuře.

Práce může rovněž posloužit pro základní pochopení práce s daty, jež jsou publikovány na Sémantickém webu v souladu s principy Linked data. V práci je shrnut celý postup vedoucí k publikaci těchto dat, od popisu formátů RDF pro zápis, editaci, až po způsoby zveřejňování propojených dat, doplněný o poměrně velké množství souvisejících zdrojů, které dané oblasti blíže specifikují.

Terminologický slovník

Termín	Zkratka	Význam [zdroj]
Common Procurement Vocabulary	CPV	Systém pro klasifikaci veřejných zakázek, standardizující odkazy, které jsou používány zadavateli a zadávajícími subjekty k popisu veřejných zakázek [38]
D2R Server		Nástroj pro publikování dat z relačních databází na Sémantickém webu [13]
Dataset		Množina dat, obvykle zobrazovaná ve formě tabulky [14]
EXtensible Markup Language	XML	Jednoduchý textový formát pro ukládání informací čitelných pro lidi i stroje [43]
File Transfer Protocol	FTP	Aplikační protokol, který slouží k přenosu souborů v počítačových sítích [24]
Framework		Softwarová struktura zahrnující podpůrné programy, knihovny či postupy, která slouží k podpoře vývoje aplikací či jiných softwarových projektů [40]
Friend Of a Friend	FOAF	Projekt věnovaný propojování informací o lidech na Webu [7]
HyperText Transfer Protocol	HTTP	Aplikační protokol, který slouží pro vyhledávání informací na Internetu [24]
Linked Data		Pojem používaný k popisu nejlepších způsobů, jak publikovat, sdílet a propojovat data na Sémantickém webu za použití URI a RDF [14]
Literál		Pojem literál představuje konstantu, která je zapsána vlastní hodnotou. Existují například číselné nebo řetězcové literály [31]
Metadata		Data vysvětlující či popisující jiná data (takzvaná data o datech) [14]
Mikroformáty		Mikroformáty slouží k vkládání strojově čitelných informací do webových dokumentů [28]
Natural Language Processing	NLP	NLP je počítačové zpracování přirozeného jazyka, je to technika pro překlad počítačového a lidského jazyka [23]

Ontologie		Formální reprezentace znalostí skrze množinu pojmů (konceptů) a vztahů mezi nimi v rámci určité domény [14]
RDF Schema	RDFS	RDFS poskytuje slovník určený k datovému modelování pro RDF data [8]
RDF trojice (triple)		Základní prvek RDF grafu, tvořen trojicí subjekt-predikát-objekt [27]
Resource Description Framework	RDF	Technologie, která slouží jako základ pro několik technologií pro modelování distribuovaných znalostí [14]
Sémantický web	SW	Web dat, který lze zpracovávat přímo či nepřímo pomocí strojů, založený na formátu RDF [14]
SPARQL Protocol And RDF Query Language	SPARQL	Množina specifikací, které poskytují jazyky a protokoly pro dotazování a manipulaci s RDF grafy [41]
Taxonomie		Technika klasifikace objektů [14]
Uniform Resource Identifier	URI	Termín pro všechny typy názvů a adres odkazujících na objekty na World Wide Webu [14]
Web Application Programming Interface	Web API	Aplikační prostřední na Webu, které umožňuje jednoduchý dotazovací přístup ke strukturovaným datům skrze HTTP protokol [17]
Web Ontology Language	OWL	Jazyk pro popis komplexních o objektech, skupinách objektů a vlastnostech mezi nimi [21]

Seznam literatury

[1] AKSW. Agile Knowledge Engineering and Semantic Web. [online]. [2014] [cit. 2015-04-17]. Dostupné z: <*http://aksw.org/About.html*>

[2] AKSW. XTurtle. Agile Knowledge Engineering and Semantic Web. [online]. [2015] [cit. 2015-04-16]. Dostupné z: <*http://aksw.org/Projects/Xturtle.html*>

[3] ALLEMANG, D. – HENDLER, J. *Semantic Web for the working ontologist: effective modeling in RDFS and OWL*. 2nd ed. Waltham, MA: Morgan Kaufmann/Elsevier, c2011, xiii, 354 p. ISBN 0123859654.

[4] BERNERS-LEE, T. Linked Data. Design Issues. [online]. 18. 6. 2009 [cit. 2015-04-01]. Dostupné z: <*http://www.w3.org/DesignIssues/LinkedData.html*>

[5] BIZER, C – HEATH, T. – BERNERS-LEE, T. Linked Data – The Story So Far [online PDF dokument]. [2009] [Cit. 2015-03-22]. Dostupný z WWW: <http://tomheath.com/papers/bizer-heath-berners-lee-ijswis-linked-data.pdf >

[6] BIZER, C. – CYGANIAK, R., HEATH, T. *How to Publish Linked Data on the Web*. [online]. 2008-07-17 [cit. 2015-04-16]. Dostupné z: <*http://sites.wiwiss.fu-berlin.de/suhl/bizer/pub/LinkedDataTutorial/20070727/*>

[7] BRICKLEY, D – Miller, L. FOAF Vocabulary Specification 0.99. [online]. [2014-01-14] [cit. 2015-04-23]. Dostupné z: <*http://xmlns.com/foaf/spec/*>

[8] BRICKLEY, D. – GUHA, R. V. RDF Schema 1.1. *W3C Recommendation.* [online]. 25. 2. 2014 [cit. 2015-04-10]. Dostupné z: <*http://www.w3.org/TR/rdf-schema/*>

[9] Cambridge Semantics. RDFS Introduction. *Cambridge Semantics.* [online]. [2015] [cit. 2015-04-10]. Dostupné z: <*http://www.cambridgesemantics.com/semantic-university/rdfs-introduction*>

[10] CAPADISLI, S. – AUER, S. – RIEDL, R. Linked Statistical Data Analysis. [online]. [2013] [cit. 2015-04-15]. Dostupné z: <*http://challenge.semanticweb.org/2013/submissions/swc2013_submission_7.pdf*>

[11] CYGANIAK, R. – JENTZSCH, A.. The Linking Open Data cloud diagram. *Design Issues.* [online]. 2014-08-30 [cit. 2015-04-05]. Dostupné z: <*http://lod-cloud.net/*>

[12] CYGANIAK, R. – REYNOLDS, D. The RDF Data Cube Vocabulary. *W3C Recommendation* . [online]. 2014-01-16 [cit. 2015-04-17]. Dostupné z: <*http://www.w3.org/TR/vocab-data-cube/*>

[13] CYGANIAK, R. Accessing Relational Databases as Virtual RDF Graphs. *The D2RQ Platform*. [online]. [2012] [cit. 2015-04-10]. Dostupné z: <*http://d2rq.org/*>

[14] data.gov.uk. Glossary of Public Sector Information and Open Data Terminology. *data.gov.uk.* [online]. [cit. 2015-04-23]. Dostupné z: <*http://data.gov.uk/glossary*>

[15] GUDGIN, M. – HADLEY, M. Web Services Addressing 1.0 – Core. *W3C Working Draft.* [online]. 2005-02-15 [cit. 2015-04-16]. Dostupné z: <*http://www.w3.org/TR/2005/WD-ws-addr-core-20050215/*>

[16] HARRIS, S. – SEABORNE, A. SPARQL 1.1 Query Language. *W3C Recommendation*. [online]. 2013-03-21 [cit. 2015-04-16]. Dostupné z: <http://www.w3.org/TR/sparql11-query/>

[17] HEATH, T. – BIZER, C. *Linked data: Evolving the web into a global data space.* [San Rafael]: Morgan & Claypool Publishers, 2011. ISBN 978-1-60845-430-3.

[18] HEATH, T.. Tools. Linked Data. [online]. [cit. 2015-04-15]. Dostupné z: <http://linkeddata.org/tools>

[19] HELMICH, J. Payola User Guide. GitHub. [online]. [2013-07-28] [cit. 2015-04-17]. Dostupné z: <https://github.com/payola/Payola/blob/master/docs/user_guide.md>

[20] HERMAN, I. – ADIDA, B – SPORNY, M. RDFa 1.1 Primer – Third Edition. W3C Working Group Note. [online]. 2015-03-17 [cit. 2015-04-10]. Dostupné z: <http://www.w3.org/TR/xhtml-rdfa-primer/>

[21] HITZLER, P. – KRÖTZSCH, M. – PARSIA, B. OWL 2 Web Ontology Language Primer (Second Edition). *W3C Recommendation.* [online]. 11. 12. 2012 [cit. 2015-04-15]. Dostupné z: <http://www.w3.org/TR/owl2-primer/>

[22] ISA-tools. stato-ontology.org. *STATO.* [online]. [2014] [cit. 2015-04-26]. Dostupné z: < http://stato-ontology.org>

[23] JANNSEN, C. Natural Language Processing (NLP). Technopedia. [online]. [2015] [cit. 2015-04-17]. Dostupné z: <http://www.techopedia.com/definition/653/natural-language-processing-nlp>

[24] KABELOVÁ, A. – DOSTÁLEK, L. *Velký průvodce protokoly TCP/IP a systémem DNS*. 5., aktualiz. vyd. Brno: Computer Press, 2008, 488 s. ISBN 978-80-251-2236-5.

[25] KALTENBÖCK, M. – BAUER, F. *Linked open data: the essentials: a quick start guide for decision makers*. Wien: ed. mono/monochrom, 2012. ISBN 9783902796059.

[26] MACKWORTH, A. – POOLE, D. Description Logic. Artificial Intelligence: Foundations of Computational Agents. [online]. [2010] [cit. 2015-04-15]. Dostupné z: <http://artint.info/html/ArtInt_317.html>

[27] MAŘÍK, V. – ŠTĚPÁNKOVÁ, O. – LAŽANSKÝ, J. *Umělá inteligence*. Vyd. 1. Praha: Academia, 2013, 489 s. ISBN 978-80-200-2276-9.

[28] Microformats community. Introduction to Microformats. *Microformats.* [online]. 2014-04-07 [cit. 2015-04-25]. Dostupné z: <http://microformats.org/wiki/introduction>

[29] Ontotext. The Truth About Triplestores. Ontotext. [online]. [2014] [cit. 2015-04-15]. Dostupné z: <http://www.ontotext.com/documents/white_papers/The-Truth-About-Triplestores.pdf>

[30] PAPASTEFANATOS, G. – PETROU, I. Publishing Statistical Data as Linked Data. [online]. [2013] [cit. 2015-04-17]. Dostupné z: <https://web.imis.athena-innovation.gr/redmine/attachments/download/1018/Data%20Cube%20Vocabulary%20Overview%20and%20Example.pdf>

[31] PECINOVSKÝ, Rudolf. *Myslíme objektově v jazyku Java: kompletní učebnice pro začátečníky*. 2., aktualiz. a rozš. vyd. Praha: Grada, 2009, 570 s. ISBN 978-80-247-2653-3.

[32] PSNET. The Public Spending ontology. [online]. [2013] [cit. 2015-04-17]. Dostupné z: <https://docs.google.com/document/d/16fxFgtjRZC5AU00RiR0jdzbRFU73cBcOGl8ZZECwl6U/edit>

[33] PUBLICSPENDING.NET. How it works. Publicspending.net. [online]. [2015] [cit. 2015-04-22]. Dostupné z: <http://publicspending.net/howitworks>

[34] PUBLICSPENDING.NET. The Public Spending ontology (PSNET). *Publicspending.net.* [online]. 2013 [cit. 2015-04-25]. Dostupné z: https://docs.google.com/document/d/16fxFgtjRZC5AU00RiR0jdzbRFU73cBcOGl8ZZECwl6U/edits

[35] REYNOLDS, D. Guide to the Payments Ontology. *data.gov.uk.* [online]. 2010-10-14 [cit. 2015-04-25]. Dostupné z: http://data.gov.uk/resources/payments

[36] Semanticweb.org. SPARQL endpoint. Semantic Web. [online]. 2011-11-26 [cit. 2015-04-16]. Dostupné z: <http://semanticweb.org/wiki/SPARQL_endpoint>

[37] SCHREIBER, G. – RAIMOND, Y. RDF 1.1 Primer. *W3C Working Group Note.* [online]. 25. 2. 2014 [cit. 2015-04-10]. Dostupné z: < http://www.w3.org/TR/2014/NOTE-rdf11-primer-20140225>

[38] Simap. CPV. Simap – Informace o evropských veřejných zakázkách. [online]. [2015] [cit. 2015-04-17]. Dostupné z: <http://simap.europa.eu/codes-and-nomenclatures/codes-cpv/codes-cpv_cs.htm>

[39] STEPHENS, Ryan K, Ronald R PLEW a Arie JONES. *Naučte se SQL za 28 dní*. Vyd. 1. Překlad Lukáš Krejčí. Brno: Computer Press, 2010, 728 s. ISBN 9788025127001.

[40] TechTerms.com. Framework. *W3C.* [online]. [2015] [cit. 2015-04-23]. Dostupné z: <http://techterms.com/definition/framework>

[41] The W3C SPARQL Working Group. SPARQL 1.1 Overview. W3C Recommendation. [online]. 2013-03-21 [cit. 2015-04-16]. Dostupné z: <http://www.w3.org/TR/2013/REC-sparql11-overview-20130321/ >

[42] UNICODE, Inc. What is Unicode?. The Unicode Consortium. [online]. [16. 1. 2015] [cit. 2015-03-22]. Dostupné z:<http://www.unicode.org/standard/WhatIsUnicode.html>

[43] W3C. Extensible Markup Language. *W3C.* [online]. [2003] [cit. 2015-04-23]. Dostupné z: <http://www.w3.org/XML/>

[44] W3C. Ontology editors. *W3C Wiki. [online]. 2015-02-02 [cit. 2015-04-10]. Dostupné z:* <http://www.w3.org/wiki/Ontology_editors>

[45] W3C. RIF FAQ. *RIF.* [online]. 4. 2. 2013 [cit. 1970-01-01]. Dostupné z: <http://www.w3.org/2005/rules/wiki/RIF_FAQ#What_is_RIF.3F>

[46] WONDERLICH, J. Ten Principles for Opening Up Government Information. *Sunlight Foundation.* [online]. 2010-08-11 [cit. 2015-04-08]. Dostupné z: <http://sunlightfoundation.com/policy/documents/ten-open-data-principles/>

Seznam obrázků a tabulek

Základní text

Seznam obrázků

Obrázek 1: Architektura Sémantického webu ([27], s. 87) 5
Obrázek 2: Linking Open Data Cloud Diagram [11] 11
Obrázek 3: Vztahy mezi třídami definovanými v RDFS [8] 16
Obrázek 4: Uživatelské rozhraní programu XTurtle 19
Obrázek 5: Uživatelské rozhraní programu Protégé 20
Obrázek 6: Způsoby publikování propojených dat [17] 22
Obrázek 7: Architektura D2R serveru [13] 25
Obrázek 8: Architektura Data Cube Vocabulary [12] 29
Obrázek 9: Architektura Payments Ontology [35] 34
Obrázek 10: Proces zpracování dat pomocí Public Spending [33] 38
Obrázek 11 Schéma tříd definovaných ontologií Public Spending [32] 39
Obrázek 12: Struktura ontologie Public Spending Data Cube 40
Obrázek 13: Vytváření úložiště v prostředí Sesame 50
Obrázek 14: Výsledek SPARQL dotazu v prostředí Sesame 51
Obrázek 15: Vizualizace dat v nástroji CubeViz 54

Seznam tabulek

Tabulka 1: Definiční obory a obory hodnot vybraných vlastností definovaných v RDF Schema [8] 16
Tabulka 2: Pojmy definované slovníkem Dublin Core, které je vhodné použít pro popis datasetu [12] 31
Tabulka 3: Státní útvary zapojené do projektu Public Spending spolu s údaji o počtu zaznamenaných plateb a celkové částce k 18. 4. 2015. 38
Tabulka 4: Přehled vlastností používaných pro návrh datové struktury [12] 42
Tabulka 5: Konkrétní data definovaná pomocí Public Spending 47

Rejstřík

— C —

Content negotiation, 7
Content-oriented Guidelines, 28, 33
CPV, 39, 40

— D —

D2R server, 21, 24
Data Cube Vocabulary, 27
Dataset, 8, 20, 30, 32

— F —

FOAF, 20, 28
Framework, 3, 27
FTP, 4

— H —

HTTP, 2

— L —

Linked Data, 3, 6
Linked Open Data, 9
Literál, 5

— M —

Metadata, 11, 28, 30, 32
Mikroformáty, 2

— N —

NLP, 37

— O —

Ontologie, 6, 8, 14, 17
OWL, 5

— P —

Protégé, 19

— R —

RDF triplestore, 25
RDF trojice, 4
RDF/XML, 13
RDFa, 7, 23
RDFS, 5, 14
Resource Description Framework, 12

— S —

SDMX, 27, 32
Sémantický web, 2
SPARQL, 49, 50
SPARQL endpoint, 22, 24

— T —

Taxonomie, 5, 11, 14
Turtle, 12

— U —

URI, 4, 7, 11

— W —

Web API, 2, 25
Web Ontology Language, 17

— X —

XML, 5
XTurtle, 18

Příloha A: Public Spending Data Cube

```
@prefix rdfs:           <http://www.w3.org/2000/01/rdf-schema#> .
@prefix rdf:            <http://www.w3.org/1999/02/22-rdf-syntax-ns#>  .
@prefix qb:             <http://purl.org/linked-data/cube#>     .
@prefix psnet:          <http://publicspending.net/ontology#> .
@prefix xsd:            <http://www.w3.org/2001/XMLSchema#> .
@prefix owl:            <http://www.w3.org/2002/07/owl#> .
@prefix sdmx-attribute: <http://purl.org/linked-data/sdmx/2009/attribute#> .
@prefix sdmx-measure:   <http://purl.org/linked-data/sdmx/2009/measure#> .
@prefix sdmx-dimension: <http://purl.org/linked-data/sdmx/2009/dimension#> .
@prefix psdc:           <https://raw.githubusercontent.com/lindahorakova/public-
spending-data-cube/master/psdc.ttl#>  .
@prefix skos:           <http://www.w3.org/2004/02/skos/core#> .
@prefix dc:             <http://purl.org/dc/terms/> .
@prefix foaf:           <http://xmlns.com/foaf/0.1/> .


<https://raw.githubusercontent.com/lindahorakova/public-spending-data-
cube/master/psdc.ttl#>
    rdf:type                    owl:Ontology;
    dc:creator                  "Linda Horakova"^^xsd:string;
    dc:date                     "2015-04-10"^^xsd:date;
    dc:description              "Ontology describes payments between payers and
                                    payees in public spending"@en;
    .

# Dataset

psdc:Dataset a qb:DataSet;
    rdfs:label                  "Platby Food and Environment Research Agency nad
                                25 000 liber"@cs;
    rdfs:label                  "Payments made by Food and Environment Research
                                Agency over 25 000 pounds"@en;
    dc:title                    "Payments between payers and peyess in public
                                spending"@cs;
    dc:description              "Dataset contains data about public spending
                                payments, made by Food and Environment Research
                                Agency."@en;
    rdfs:comment                "Dataset contains data about public spending
                                payments, made by Food and Environment Research
                                Agency."@en;
    dc:issued                   "2015-04-18"^^xsd:date;
    dc:modified                 "2015-04-18"^^xsd:date;
    dc:subject                  "Payments between payers and payees"^^xsd:string;
    dc:publisher                "Linda Horakova"^^xsd:string;
```

```
    qb:structure              psdc:DSD;
    .

# Data Structure Definition

psdc:DSD a qb:DataStructureDefinition;
    rdfs:label                "Definice datove struktury pro dataset - Platby
                              mezi platci a prijemci v oblasti verejnych
                              vydaju"@cs;
    rdfs:label                "Definition of data structure for dataset -
                              Payments between payers and payees in public
                              spending"@en;
    qb:component              psdc:payerComponent, psdc:payeeComponent,
                              psdc:dateComponent, psdc:cpvComponent,
                              psdc:AmountComponent,psdc:currencyComponent;
    .


psdc:payerComponent a qb:ComponentSpecification;
    rdfs:label                "Specification of the component describing
                              a payer"@en;
    rdfs:label                "Specifikace komponenty popisujici platce"@cs;
    qb:dimension              psdc:payer;
    qb:order                  1 .

psdc:payeeComponent a qb:ComponentSpecification;
    rdfs:label                "Specification of the component describing
                              a payee"@en;
    rdfs:label                "Specifikace komponenty popisujici prijemce"@cs;
    qb:dimension              psdc:payee;
    qb:order                  2 .

psdc:dateComponent a qb:ComponentSpecification;
    rdfs:label                "Specification of the component describing a date
                              of the payment"@en;
    rdfs:label                "Specifikace komponenty popisujici datum
                              platby"@cs;
    qb:dimension              psdc:date;
    qb:order                  3 .

psdc:cpvComponent a qb:ComponentSpecification;
    rdfs:label                "Specification of the component describing a CPV
                              code"@en;
    rdfs:label                "Specifikace komponenty popisujici CPV kod"@cs;
    qb:dimension              psdc:cpv;
    qb:order                  4 .

psdc:AmountComponent a qb:ComponentSpecification;
```

```
    rdfs:label "Specification of the component describing a amount"@en;
    rdfs:label "Specifikace komponenty popisujici castku"@cs;
    qb:measure psdc:amount;
    .

psdc:currencyComponent a qb:ComponentSpecification;
    rdfs:label                "Specification of the component describing
                              a currency"@en;
    rdfs:label                "Specifikace komponenty popisujici menu"@cs;
    qb:attribute              sdmx-attribute:currency .

# Dimensions, measures

psdc:payer a rdf:Property, qb:DimensionProperty;
    rdfs:label                "Platce"@cs;
    rdfs:label                "Payer"@en;
    rdfs:subPropertyOf        psnet:payer;
    rdfs:domain               qb:Observation;
    rdfs:range                psdc:PaymentAgent;
    .

psdc:payee a rdf:Property, qb:DimensionProperty;
    rdfs:label                "Prijemce"@cs;
    rdfs:label                "Payee"@en;
    rdfs:subPropertyOf        psnet:payee;
    rdfs:domain               qb:Observation;
    rdfs:range                psdc:PaymentAgent;
    .

psdc:date a owl:DatatypeProperty, qb:DimensionProperty;
    rdfs:label                "Datum platby"@cs;
    rdfs:label                "Date"@en;
    rdfs:subPropertyOf        sdmx-dimension:refTime;
    rdfs:domain               qb:Observation;
    rdfs:range                xsd:date;
    .

psdc:cpv a owl:DatatypeProperty, qb:DimensionProperty;
    rdfs:label                "Common Procurement Vocabulary"@en;
    rdfs:label                "Spolecny slovnik pro verejne zakazky"@cs;
    rdfs:subPropertyOf        psnet:cpv;
    rdfs:domain               qb:Observation;
    qb:codeList               <http://purl.org/cpv/2008/scheme-CPV2008>;
    rdfs:range                skos:Concept;
    .


psdc:amount a rdf:Property, qb:MeasureProperty;
```

```
    rdfs:label                  "Penezni castka"@cs;
    rdfs:label                  "Amount of money"@en;
    rdfs:subPropertyOf          sdmx-measure:obsValue;
    rdfs:range                  xsd:decimal;
    .

# Classes

psdc:PaymentAgent a rdfs:Class;
    rdfs:label                  "Payment Agent"@en;
    rdfs:label                  "Ucastnik platby"@cs;
    rdfs:subClassOf             psnet:PaymentAgent;
    rdfs:subClassOf             foaf:Agent;
    .
```

Příloha B: Ukázková data pro Public Spending Data Cube

```
psdc:payment1 a qb:Observation;
 qb:dataSet                psdc:Dataset;
 psdc:amount               166052.36;
 sdmx-attribute:currency   <http://ecb.270a.info/code/1.0/CL_CURRENCY/GBP>;
 psdc:payer
 <http://publicspending.medialab.ntua.gr/resource/uk/entities/Fera>;
 psdc:payee
 <http://publicspending.medialab.ntua.gr/resource/uk/suppliers/National_Westminster_Bank_Plc>;
 psdc:date                 "2004-12-01";
 psdc:cpv                  <http://purl.org/cpv/2008/code-22452000-3>;
.

psdc:payment2 a qb:Observation;
 qb:dataSet                psdc:Dataset;
 psdc:amount               287549.39;
 sdmx-attribute:currency   <http://ecb.270a.info/code/1.0/CL_CURRENCY/GBP>;
 psdc:payer
 <http://publicspending.medialab.ntua.gr/resource/uk/entities/Fera>;
 psdc:payee
 <http://publicspending.medialab.ntua.gr/resource/uk/suppliers/Forestry_Commission>;
 psdc:date                 "2005-11-01";
 psdc:cpv                  <http://purl.org/cpv/2008/code-79342100-4>;
 .

psdc:payment3 a qb:Observation;
 qb:dataSet                psdc:Dataset;
 psdc:amount               47459.95;
 sdmx-attribute:currency   <http://ecb.270a.info/code/1.0/CL_CURRENCY/GBP>;
 psdc:payer
 <http://publicspending.medialab.ntua.gr/resource/uk/entities/Fera>;
 psdc:payee
 <http://publicspending.medialab.ntua.gr/resource/uk/suppliers/Forestry_Commission>;
 psdc:date                 "2005-07-01";
 psdc:cpv                  <http://purl.org/cpv/2008/code-79342100-4>;
 .

psdc:payment4 a qb:Observation;
 qb:dataSet                psdc:Dataset;
psdc:amount                35250.73;
sdmx-attribute:currency    <http://ecb.270a.info/code/1.0/CL_CURRENCY/GBP>;
```

```
 psdc:payer
 <http://publicspending.medialab.ntua.gr/resource/uk/entities/Fera>;
 psdc:payee
 <http://publicspending.medialab.ntua.gr/resource/uk/suppliers/Innoventors_Ltd>;
 psdc:date                "2006-05-01";
 psdc:cpv                 <http://purl.org/cpv/2008/code-79342100-4>;
 .

psdc:payment5 a qb:Observation;
 qb:dataSet               psdc:Dataset;
 psdc:amount              41178.06;
 sdmx-attribute:currency  <http://ecb.270a.info/code/1.0/CL_CURRENCY/GBP>;
 psdc:payer
 <http://publicspending.medialab.ntua.gr/resource/uk/entities/Fera>;
 psdc:payee
 <http://publicspending.medialab.ntua.gr/resource/uk/suppliers/&apos;veterinary_Laboratories_Agency>;
 psdc:date                "2005-02-01";
 psdc:cpv                 <http://purl.org/cpv/2008/code-79342100-4>;
 .

psdc:payment6 a qb:Observation;
 qb:dataSet               psdc:Dataset;
 psdc:amount              32556.72;
 sdmx-attribute:currency  <http://ecb.270a.info/code/1.0/CL_CURRENCY/GBP>;
 psdc:payer
 <http://publicspending.medialab.ntua.gr/resource/uk/entities/Fera>;
 psdc:payee
 <http://publicspending.medialab.ntua.gr/resource/uk/suppliers/Innoventors_Ltd>;
 psdc:date                "2006-05-01";
 psdc:cpv                 <http://purl.org/cpv/2008/code-79342100-4>;
 .

psdc:payment7 a qb:Observation;
 qb:dataSet               psdc:Dataset;
 psdc:amount              46053.01;
 sdmx-attribute:currency  <http://ecb.270a.info/code/1.0/CL_CURRENCY/GBP>;
 psdc:payer
 <http://publicspending.medialab.ntua.gr/resource/uk/entities/Fera>;
 psdc:payee
 <http://publicspending.medialab.ntua.gr/resource/uk/suppliers/Ni-co>;
 psdc:date                "2004-07-01";
 psdc:cpv                 <http://purl.org/cpv/2008/code-79342100-4>;
 .

psdc:payment8 a qb:Observation;
 qb:dataSet               psdc:Dataset;
 psdc:amount              52081.01;
 sdmx-attribute:currency  <http://ecb.270a.info/code/1.0/CL_CURRENCY/GBP>;
```

```
  psdc:payer
  <http://publicspending.medialab.ntua.gr/resource/uk/entities/Fera>;
  psdc:payee
  <http://publicspending.medialab.ntua.gr/resource/uk/suppliers/Ip_Pragmatics_Limited>;
  psdc:date                 "2004-09-01";
  psdc:cpv                  <http://purl.org/cpv/2008/code-79342100-4>;
.

psdc:payment9 a qb:Observation;
  qb:dataSet                psdc:Dataset;
  psdc:amount               25000.0;
  sdmx-attribute:currency   <http://ecb.270a.info/code/1.0/CL_CURRENCY/GBP>;
  psdc:payer
  <http://publicspending.medialab.ntua.gr/resource/uk/entities/Fera>;
  psdc:payee
  <http://publicspending.medialab.ntua.gr/resource/uk/suppliers/University_Of_Nottingham>;
  psdc:date                 "2005-02-01";
  psdc:cpv                  <http://purl.org/cpv/2008/code-79342100-4>;
  .
```